Orthogonal Polynomials and Symbolic Computation

V. Pillwein

January 20, 2012

Contents

Chapter 1

Definition and general facts

1.1 Literature and conventions

Some classical literature on orthogonal polynomials and special functions:

- Andrews, Askey, Roy: "Special Functions" [2]
- Chihara: "An introduction to orthogonal polynomials" [4]
- Rainville: "Special Functions" [17]
- Szegö: "Orthogonal Polynomials" [19]

We will use the following conventions during this lecture:

- throughout, let K be a field containing the rational numbers, i.e., a field of characteristic zero; usually $K = \mathbb{R}(\mathbb{C})$;
- $\mathbb{N} = \{0, 1, 2, \dots\}$;
- $K[x]$ denotes the ring of polynomials in the indeterminate x with coefficients in K;
- the coefficient functional is denoted by $\langle x^m \rangle \sum_{k=0}^{n} a_k x^k := a_m$.

1.2 Normal sequences

Before we turn to orthogonal polynomials we define another notion for polynomial bases.

Definition 1.1. *A sequence of polynomials $(\phi_n(x))_{n\geq 0}$ is called* normal*, iff for all $n \in \mathbb{N}$: $\deg \phi_n(x) = n$.*

These polynomials are sometimes also called *simple* and by definition in a sequence of normal polynomials there is exactly one polynomial of degree n for each $n \in \mathbb{N}$. From this definition it is immediate that any polynomial can be expressed as linear combination of elements of a normal sequence.

Theorem 1.2. *Any normal sequence $(\phi_n(x))_{n\geq 0}$ forms a basis of $K[x]$ viewed as vector space over K.*

Proof. Let $p \in K[x]$ be a polynomial of degree m, i.e., the leading coefficient $\langle x^m \rangle p(x) = a_m \neq 0$. Since $\phi_m(x)$ is normal, also its leading coefficient b_m does not vanish for any m. Define $c_m = \frac{a_m}{b_m}$ and $p^{(1)}(x) = p(x) - c_m \phi_m(x)$. Then we have a reduction in the polynomial degree, i.e., $\deg p^{(1)}(x) \leq m-1$. Applying this procedure iteratively yields a unique expansion $p(x) = \sum_{k=0}^{m} c_k \phi_k(x)$, where not all c_k are zero. □

Some examples for normal sequences are:

- monomial basis: $(x^n)_{n \geq 0} = \big(1, x, x^2, \dots\big)$
- falling factorial: $(x^{\underline{n}})_{n \geq 0}$, where

$$x^{\underline{n}} := \begin{cases} x \cdot (x-1) \cdots (x-n+1) & n \geq 1, \\ 1 & n = 0. \end{cases}$$

- rising factorial: $(x^{\bar{n}})_{n \geq 0}$, where

$$x^{\bar{n}} := \begin{cases} x \cdot (x+1) \cdots (x+n-1) & n \geq 1, \\ 1 & n = 0. \end{cases}$$

Often the rising factorial is also denoted using the Pochhammer symbol $x^{\bar{n}} = (x)_n$.

By theorem 1.2 we can convert one basis (normal sequence) into any other. We consider as example the conversion from the (classical) monomial basis to falling factorials, e.g., determine c_k such that

$$p(x) = x^3 = \sum_{k=0}^{3} a_k x^k = \sum_{k=0}^{3} c_k x^{\underline{k}},$$

where $a_0 = a_1 = a_2 = 0$ and $a_3 = 1$. The falling factorials we need are:

$$x^{\underline{0}} = 1, \quad x^{\underline{1}} = x, \quad x^{\underline{2}} = x^2 - x, \quad \text{and} \quad x^{\underline{3}} = x^3 - 3x^2 + 2x.$$

Using the notation from the proof of theorem 1.2 we calculate:

$$\begin{aligned} a_3 = b_3 = 1 \quad & \Rightarrow c_3 = 1: \quad & p^{(1)}(x) = p(x) - 1 \cdot x^{\underline{3}} = 3x^2 - 2x \\ a_2 = 3,\ b_2 = 1 \quad & \Rightarrow c_2 = 3: \quad & p^{(2)}(x) = p^{(1)}(x) - 3x^{\underline{2}} = x \\ a_1 = b_1 = 1 \quad & \Rightarrow c_1 = 1: \quad & p^{(3)}(x) = p^{(2)}(x) - x^{\underline{1}} = 0. \end{aligned}$$

Thus $c_0 = 0$ and so we have $x^3 = x^{\underline{3}} + 3x^{\underline{2}} + x^{\underline{1}}$.

Remark 1.3. *The transfer or connection coefficients for expressing the monomial basis in terms of the falling factorials are the Stirling numbers of the 2nd kind $S(n,k)$ that have the combinatorial definiton*

$S(n,k) =$ *number of set partitions of an $n-$ element set into exactly k nonempty, disjoint subsets.*

It holds that $x^n = \sum_{k=0}^{n} S(n,k) x^{\underline{k}}$.

1.3 Orthogonality

If we want to discuss orthogonality, we need the notion of an inner product that we recall in the following definition for the real case.

Definition 1.4. *Let V be a real vector space. A map $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ is called* inner product*, iff it is*

1. *positive definite, i.e.,*

$$(\forall\, f \in V : \langle f, f \rangle \geq 0) \quad \text{and} \quad (\langle f, f \rangle = 0 \Leftrightarrow f = 0)\,,$$

2. *symmetric, i.e.,*

$$\forall\, f, g \in V : \langle f, g \rangle = \langle g, f \rangle,$$

3. *bilinear, i.e.,*

$$\forall\, f, g \in V\ \forall \lambda, \mu \in \mathbb{R} : \langle \lambda f + \mu g, h \rangle = \lambda \langle f, h \rangle + \mu \langle g, h \rangle.$$

In this lecture we are dealing with special types of inner products that are defined via weight functions. Let $I = (a, b)$ be a real interval with $a, b \in \mathbb{R} \cup \{\pm\infty\}$ that can be either open, half-open or closed. Let $w(x)$ be a function defined on (a, b) with $w(x) > 0$. Then

$$\langle f, g \rangle_w := \int_a^b f(x) g(x) w(x)\ \mathrm{dx} \tag{1.1}$$

defines an inner product and $w(x)$ is called a *weight function*. We denote the space of weighted square integrable functions over an interval I by $L^2_w(I)$, i.e.,

$$L^2_w(I) = \left\{ f : I \to \mathbb{R}\ :\ \int_I f(x)^2 w(x)\ \mathrm{dx} < \infty \right\}.$$

For sake of simplicity from now on we will always assume that the moments of all orders exist, i.e.,

$$\int_a^b x^n w(x)\ \mathrm{dx} < \infty, \qquad n \in \mathbb{N}.$$

In this case $w(x)$ is called a proper weight function.

Definition 1.5. *Let $(\phi_n(x))_{n\geq 0}$, $\phi_n(x) \in \mathbb{R}[x]$, be a normal sequence and let $w(x)$ be a proper weight function on the interval $I = (a, b)$. If*

$$\langle \phi_n, \phi_m \rangle_w = \int_a^b \phi_n(x) \phi_m(x) w(x)\ \mathrm{dx} = 0 \qquad \text{for } n \neq m,$$

we say that $(\phi_n(x))_{n\geq 0}$ is orthogonal with respect to the weight function $w(x)$ on I.

None of the normal sequences that we have seen so far are orthogonal with respect to any weight function. Some famous examples for orthogonal polynomials that we will discuss in more detail later are

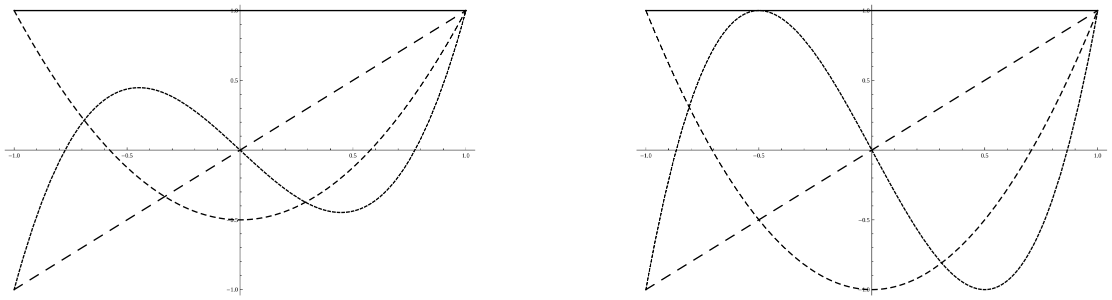


Figure 1.1: Legendre polynomials $P_n(x)$ (left) and Chebyshev polynomials $T_n(x)$ (right)

- Legendre polynomials $P_n(x)$ are orthogonal with respect to the constant weight function $w(x) \equiv 1$ on the interval $[-1, 1]$. The first few are given by

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = \tfrac{1}{2}(3x^2 - 1), \quad P_3(x) = \tfrac{1}{2}x(5x^2 - 3).$$

- Chebyshev polynomials of the 1st kind $T_n(x)$ are orthogonal with respect to the weight function $w(x) = \sqrt{1 - x^2}$ on the interval $[-1, 1]$. The first few are given by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = x(4x^2 - 3).$$

The next theorem gives an alternative characterization of orthogonal polynomials.

Theorem 1.6. *Let $(\phi_n(x))_{n \geq 0}$ be a normal sequence and $w(x)$ be a proper weight function on (a, b). Then $\phi_n(x)$ is orthogonal with respect to $w(x)$ if and only if*

$$\forall\, n \in \mathbb{N} : \qquad \int_a^b x^k \phi_n(x) w(x) \,\mathrm{dx} = 0, \qquad 0 \leq k \leq n - 1.$$

Proof. Assume that $\phi_n(x)$ is a sequence of orthogonal polynomials. Then by theorem 1.2 for each k we can compute connection coefficients $b(j, k)$ and write $x^k = \sum_{j=0}^{k} b(j, k)\phi_j(x)$. Thus

$$\int_a^b x^k \phi_n(x) w(x) \,\mathrm{dx} = \sum_{j=0}^{k} b(j, k) \int_a^b \phi_j(x)\phi_n(x) w(x) \,\mathrm{dx}.$$

But the latter integral vanishes if $j < n$ by orthogonality. The other direction is immediate. □

A simple consequence of this result is

Corollary 1.7. *Let ϕ_n, w be as in theorem 1.6. For any $p \in \mathbb{R}[x]$ with $\deg p(x) < n$ we have $\int_a^b p(x)\phi_n(x)w(x) \,\mathrm{dx} = 0$, and in addition $\int_a^b x^n \phi_n(x) w(x) \,\mathrm{dx} \neq 0$.*

Proof. Exercise. □

1.4 Three term recurrence

Throughout we will sometimes denote the squared, weighted L^2-norm of a sequence of orthogonal polynomials by

$$h_n = \langle \phi_n, \phi_n \rangle = \int_a^b \phi_n(x)^2 w(x) \, \mathrm{dx},$$

and the leading coefficient by

$$\alpha_n = \langle x^n \rangle \phi_n(x) = \mathrm{lc}\left(\phi_n(x)\right).$$

An orthogonal sequence with $h_n = 1$ for all $n \in \mathbb{N}$ is called *orthonormal*.

Theorem 1.8. *(Three term recurrence) Let $(\phi_n(x))_{n\geq 0}$ be a sequence of polynomials orthogonal with respect to $w(x)$ on (a,b). Then there exist sequences of numbers a_n, b_n, c_n, $n \in \mathbb{N}$, s.t.*

$$\phi_{n+1}(x) = \left(a_n x + b_n\right) \phi_n(x) + c_n \phi_{n-1}(x), \tag{1.2}$$

for $n \geq 0$, where we put $\phi_{-1}(x) = 0$. Moreover, we have for $n \geq 0$:

$$a_n = \frac{\alpha_{n+1}}{\alpha_n}, \quad \text{and} \quad c_{n+1} = -\frac{a_{n+1}}{a_n} \frac{h_{n+1}}{h_n}. \tag{1.3}$$

Proof. We start by proving the existence of a three term recurrence and for this we expand $x\phi_n(x)$ in terms of the basis $\phi_k(x)$. Since $x\phi_n(x)$ is a polynomial of degree $n+1$ we need an expansion in $\phi_0(x), \ldots, \phi_{n+1}(x)$:

$$x\phi_n(x) = \sum_{k=0}^{n+1} \gamma(n,k)\phi_k(x).$$

We multiply the above equation by $w(x)\phi_j(x)$ $(0 \leq j \leq n+1)$ and integrate over (a,b):

$$\int_a^b x\phi_n(x)\phi_j(x)w(x) \, \mathrm{dx} = \sum_{k=0}^{n+1} \gamma(n,k) \underbrace{\int_a^b \phi_k(x)\phi_j(x)w(x) \, \mathrm{dx}}_{=\delta_{jk}h_j}.$$

By corollary 1.7 the integral on the left hand side is nonzero only if $j = n-1, n, n+1$. This yields

$$x\phi_n(x) = \gamma(n,n-1)\phi_{n-1}(x) + \gamma(n,n)\phi_n(x) + \gamma(n,n+1)\phi_{n+1}(x)$$

and thus the existence of (1.2). Note that furthermore we have that

$$\gamma(n,j) = \frac{1}{h_j}\int_a^b x\phi_n(x)\phi_j(x)w(x) \, \mathrm{dx}, \qquad j = n-1, n, n+1.$$

For (1.3) we consider the three term recurrence in the form

$$\phi_{n+1}(x) - a_n x \phi_n(x) = b_n \phi_n(x) + c_n \phi_{n-1}(x).$$

On the right hand side we have a polynomial of degree n. Thus on the left hand side the leading coefficients must cancel and so $a_n = \alpha_{n+1}/\alpha_n$. If we multiply the above equation by $\phi_{n-1}(x)w(x)$ and integrate over (a,b), then by orthogonality we obtain

$$-a_n \underbrace{\int_a^b \phi_n(x)x\phi_{n-1}(x)w(x)\,\mathrm{dx}}_{=\gamma(n-1,n)h_n} = c_n \underbrace{\int_a^b \phi_{n-1}(x)^2 w(x)\,\mathrm{dx}}_{=h_{n-1}}.$$

Summarizing this yields $c_n = -\frac{a_n}{a_{n-1}}\frac{h_n}{h_{n-1}}$ as claimed. □

Remark 1.9.

- *The converse to theorem 1.8 also holds (under some restrictions) and this result is due to Favard (1935) [7], however the proof is not constructive and the weight function need not be a "regular" function.*
- *Dickinson, Pollak and Wannier (1956) [5] show how to construct the weight function from the three term recurrence in certain special cases.*
- *Note that theorem 1.8 yields a recurrence and a closed form for h_n given the recurrence coefficients of the given sequence of orthogonal polynomials:*

$$\begin{aligned} h_n &= -\frac{a_{n-1}}{a_n} c_n h_{n-1} \\ &= +\frac{a_{n-1}}{a_n}\frac{a_{n-2}}{a_{n-1}} c_n c_{n-1} h_{n-2} \\ &= \dots \\ &= (-1)^n \frac{a_0}{a_n} c_n c_{n-1}\cdots c_1 h_0. \end{aligned} \tag{1.4}$$

Theorem 1.10. *(Christoffel-Darboux) Let $(\phi_n(x))_{n\geq 0}$ be a sequence of orthogonal polynomials. Then*

$$\sum_{k=0}^{n} \frac{1}{h_k}\phi_k(x)\phi_k(y) = \frac{1}{a_n h_n}\frac{\phi_{n+1}(x)\phi_n(y) - \phi_n(x)\phi_{n+1}(y)}{x-y}. \tag{1.5}$$

Proof. We start by considering the right hand side of (1.5). Using the three term recurrence for $\phi_{k+1}(x)$ and $\phi_{k+1}(y)$, respectively, we obtain

$$\begin{aligned} \phi_{k+1}(x)\phi_k(y) &= (a_k x + b_k)\phi_k(x)\phi_k(y) + c_k\phi_{k-1}(x)\phi_k(y) \\ \phi_k(x)\phi_{k+1}(y) &= (a_k y + b_k)\phi_k(x)\phi_k(y) + c_k\phi_k(x)\phi_{k-1}(y) \end{aligned}$$

Subtracting these two lines gives

$$\phi_{k+1}(x)\phi_k(y) - \phi_k(x)\phi_{k+1}(y) = a_k(x-y)\phi_k(x)\phi_k(y) + c_k\left(\phi_{k-1}(x)\phi_k(y) - \phi_k(x)\phi_{k-1}(y)\right).$$

In the next step we divide this equation by $a_k h_k$ and use that $c_k = -\frac{a_k}{a_{k-1}}\frac{h_k}{h_{k-1}}$ to obtain

$$\begin{aligned} \underbrace{\frac{1}{a_k h_k}\left(\phi_{k+1}(x)\phi_k(y) - \phi_k(x)\phi_{k+1}(y)\right)}_{=:\psi_k(x,y)} &= \frac{1}{h_k}(x-y)\phi_k(x)\phi_k(y) \\ &\quad - \underbrace{\frac{1}{a_{k-1}h_{k-1}}\left(\phi_{k-1}(x)\phi_k(y) - \phi_k(x)\phi_{k-1}(y)\right)}_{=\psi_{k-1}(x,y)}. \end{aligned}$$

We rewrite the above and sum over $k = 0, \dots, n$:

$$\sum_{k=0}^{n} (\psi_k(x,y) - \psi_{k-1}(x,y)) = (x-y) \sum_{k=0}^{n} \frac{1}{h_k} \phi_k(x)\phi_k(y).$$

The left hand side telescopes to $\psi_n(x,y)$ (using $\psi_{-1}(x,y) = 0$) and since $a_n = \alpha_{n+1}/\alpha_n$ we have thus completed the proof. □

The polynomials defined in theorem 1.10 are also called *kernel polynomials* and often denoted by

$$k_n(x,y) = \sum_{j=0}^{n} \frac{1}{h_j} \phi_j(x)\phi_j(y).$$

It can be shown that for α with $-\infty < \alpha \leq a < \infty$ the sequence $(k_n(x,\alpha))_{n\geq 0}$ is orthogonal with respect to the weight function $(x-\alpha)w(x)$. From the closed form expression of $k_n(x,y)$ the following theorem is immediate.

Theorem 1.11.

$$\sum_{j=0}^{n} \frac{1}{h_j} \phi_j(x)^2 = \frac{1}{a_n h_n} \left(\phi'_{n+1}(x)\phi_n(x) - \phi'_n(x)\phi_{n+1}(x) \right).$$

Proof. Insert $\pm\phi_{n+1}(y)\phi_n(y)$ on the right hand side in (1.5), regroup terms and pass to the limit $y \to x$. □

1.5 Zeros of orthogonal polynomials

The last, but not least, of the surprising properties that we discuss here and that hold in general for all families of orthogonal polynomials concern the roots of orthogonal polynomials.

Theorem 1.12. *Let $(\phi_n(x))_{n\geq 0}$, $\phi_n \in \mathbb{R}[x]$, be a sequence of orthogonal polynomials. Then the zeros of $\phi_n(x)$ are all distinct and lie in (a,b).*

Proof. For $n = 0$ nothing needs to be proven. So let $n > 0$. Then by orthogonality

$$\int_a^b \phi_n(x)w(x) \,\mathrm{dx} = 0.$$

This means that $\phi_n(x)$ has to change signs in (a,b) at least once (recall that $w(x) > 0$). Suppose it changes signs at $x_1 < x_2 < \cdots < x_s \in (a,b)$. Obiviously the multiplicities of x_i have to be odd for $1 \leq i \leq s$. Since $\deg \phi_n(x) = n$, we know that $s \leq n$ and we want to show that in fact $s = n$.

Hence we assume that $s < n$ and define $\psi_s(x) = (x-x_1)\cdots(x-x_s)$. By corollary 1.7:

$$\int_a^b \psi_s(x)\phi_n(x)w(x) \,\mathrm{dx} = 0.$$

But then $\psi_s(x)\phi_n(x)$ has to change sign in (a,b), which cannot happen since $\psi_s(x)$ and $\phi_n(x)$ change sign in the exact same places. Thus $s = n$. □

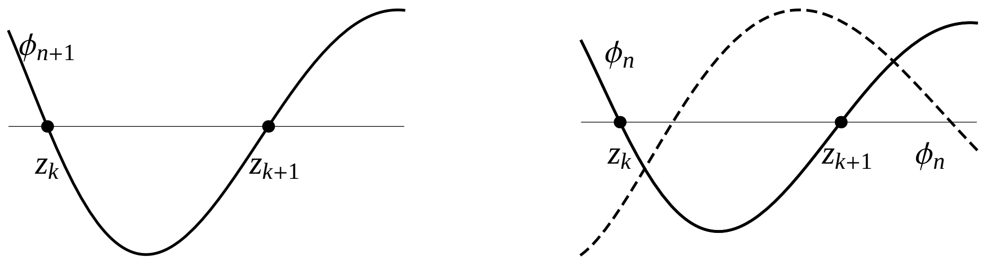


Figure 1.2: $\phi'_{n+1}(z_k)\phi_n(z_k) > 0$: case with $\phi'_{n+1}(z_k) < 0$ and $\phi_n(z_k) < 0$

This, together with the closed form given in theorem 1.10, yields the following result:

Theorem 1.13. *Let $\phi_n(x)$ be as before. The zeros of $\phi_n(x)$ and $\phi_{n+1}(x)$ separate each other.*

Proof. Recall the closed form from theorem 1.11:

$$\sum_{j=0}^{n} \frac{1}{h_j}\phi_j(x)^2 = \frac{1}{a_n h_n}\left(\phi'_{n+1}(x)\phi_n(x) - \phi'_n(x)\phi_{n+1}(x)\right).$$

From the left hand side of the identity it is obvious that the above is strictly greater 0. Assume that $a_n > 0$, then

$$\phi'_{n+1}(x)\phi_n(x) - \phi'_n(x)\phi_{n+1}(x) > 0 \tag{1.6}$$

Let $z_1 < z_2 < \cdots < z_{n+1}$ denote the (simple) roots of $\phi_{n+1}(x)$ and plug $x = z_k$ into (1.6) giving $\phi'_{n+1}(z_k)\phi_n(z_k) > 0$. This leaves two possible scenarios, one of which we depict in Figure 1.2. In this case $\phi'_{n+1}(z_k) < 0$ and $\phi'_{n+1}(z_{k+1}) > 0$. Since we have only simple roots, $\phi'_{n+1}(x)$ has to change signs in (z_k, z_{k+1}). If we plug in $x = z_{k+1}$ this yields $\phi'_{n+1}(z_{k+1})\phi_n(z_{k+1}) > 0$. Thus also ϕ_n has to change signs in (z_k, z_{k+1}). The missing case distinctions are argued analogously. □

1.6 Gauß quadrature

Often and since long ago the need arises to approximate a definite integral that cannot be evaluated exactly. Newton used the method of interpolating a function at n points and integrating the interpolating function, which supposedly should be an easy-to-evaluate integral. For instance, if the given function is approximated by a piecewise linear function we end up with the trapezoidal rule. In the following we discuss a method that uses an approximation by polynomials of higher degree. For this we introduce the Lagrange interpolation polynomial.

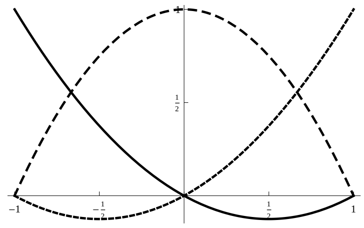
The goal is to construct a polynomial of degree $n-1$ that thake the given values $y_1, \ldots, y_n$ at the given points $x_1 < x_2 < \cdots < x_n$. First we define the auxiliary polynomial

$$P(x) = \prod_{j=1}^{n}(x - x_j) \qquad \left(\Rightarrow \quad P'(x) = \sum_{k=1}^{n}\prod_{j\neq k}(x - x_j)\right).$$

Using $P(x)$ we can define the following polynomial of degree $n-1$:

$$\ell_j(x) = \frac{P(x)}{P'(x_j)(x - x_j)}.$$

It is easy to see that $\ell_j(x_i) = \delta_{ij}$ and that $\ell_1(x), \ldots, \ell_n(x)$ forms a partition of unity.



Example 1.14. *Let* $x_1 = -1, x_2 = 0, x_3 = 1$*, then*

$$\ell_1(x) = \frac{1}{2}(x-1)x,$$
$$\ell_2(x) = -(x-1)(x+1),$$
$$\ell_3(x) = \frac{1}{2}x(x+1).$$

Now we are in the position to define the *Lagrange interpolation polynomial* for given interpolation points x_j and values y_j:

$$L_n(x) = \sum_{j=1}^{n} y_j \ell_j(x).$$

If we choose $y_j = f(x_j)$ then $L_n(x)$ is the polynomial of degree $n-1$ that interpolates f exactly at $x = x_j$ for $1 \le j \le n$. Following Newton's idea, we approximate the integral

$$\int_a^b f(x)w(x)\ \mathrm{dx} \approx \sum_{j=1}^{n} f(x_j) \underbrace{\int_a^b \ell_j(x)w(x)\ \mathrm{dx}}_{=:\lambda_j}. \tag{1.7}$$

The λ_j defined above are called "quadrature weights". Certainly this approximation is exact, if $f(x)$ is a polynomial of degree $n-1$. This rises the question on how to measure the quality of an approximation by a quadrature method (or numerical integration in general). One possibility is to compare how big the class of functions is for which the approximation is exact.

In (1.7) we use $2n$ parameters $(x_j, \lambda_j)_{j=1}^n$ to achieve an approximation that is exact up to degree $n-1$. So there is clearly room for improvement. One freedom that we did not exploit so far is the choice of the interpolation points x_j.

Theorem 1.15. *(Gauss quadrature) Let* ϕ_n, w *be as before and let* $x_1 < x_2 < \cdots < x_n$ *be the roots of* $\phi_n(x)$*. Then the weights*

$$\lambda_j = \int_a^b \ell_j(x)w(x)\ \mathrm{dx}$$

are positive for $1 \le j \le n$ *and for every* $f \in \mathbb{R}[x]$ *with* $\deg f(x) \le 2n-1$ *we have*

$$\int_a^b f(x)w(x)\ \mathrm{dx} = \sum_{j=1}^{n} \lambda_j f(x_j).$$

Proof. Let $f \in \mathbb{K}[x]$. By the division algorithm we can write

$$f(x) = q(x)\phi_n(x) + r(x), \qquad \text{with} \quad \deg r(x) \le n-1.$$

Then $f(x_j) = r(x_j)$ for all $1 \le j \le n$ and furthermore

$$\int_a^b f(x)w(x)\ \mathrm{dx} = \int_a^b q(x)\phi_n(x)w(x)\ \mathrm{dx} + \int_a^b r(x)w(x)\ \mathrm{dx}.$$

The first integral on the right hand side becomes zero, if $\deg q(x) \leq n-1$ by corollary 1.7, which is equivalent to the condition $\deg f(x) \leq 2n-1$. Summarizing

$$\int_a^b f(x)w(x)\,\mathrm{dx} = \int_a^b r(x)w(x)\,\mathrm{dx},$$

and for polynomials up to degree $n-1$ we know that Lagrange interpolation is exact and thus the quadrature formula is exact.

Now for the positivity of the quadrature weights. Choose $f(x) = \ell_k(x)^2$, then the degree of f is $2n-2$. Arguing as before, we can apply the division algorithm and write $f(x) = q(x)\phi_n(x) + r(x)$, with $\deg r(x) \leq n-1$. Now, since $f(x_j) = r(x_j)$ for all $1 \leq j \leq n$, we have that $r(x) = \ell_k(x)$. Hence

$$\lambda_k = \int_a^b \ell_k(x)w(x)\,\mathrm{dx} = \int_a^b r(x)w(x)\,\mathrm{dx} = \int_a^b f(x)w(x)\,\mathrm{dx} = \int_a^b \ell^2(x)w(x)\,\mathrm{dx} > 0.$$

□

Remark 1.16.

- *Gauß considered $w(x) \equiv 1$; the corresponding orthogonal polynomials are Legendre polynomials $P_n(x)$.*
- *often used are also Chebyshev polynomials of the first kind $T_n(x)$; in this case it can be shown that $\lambda_1 = \lambda_2 = \cdots = \lambda_n$ and the approximation for $f \in \mathbb{R}[x]$ reduces to*

$$\int_{-1}^1 f(x)\frac{\mathrm{dx}}{\sqrt{1-x^2}} = \frac{\pi}{n}\sum_{j=1}^n f\left(\cos\left(\frac{2j-1}{2n}\pi\right)\right).$$

- *the integration points and weights can be obtained by inserting sufficiently many linear independent functions with known integrals and solving the resulting system of nonlinear, algebraic equations.*
- *the quadrature formula $q_n(f) = \sum_{j=1}^n \lambda_j f(x_j)$ converges if*

  1.
$$\lim_{n->\infty} q_n(p) = \int_a^b p(x)\,\mathrm{dx}, \qquad \forall\, p \in \mathbb{R}[x],$$

  2.
$$\sum_{j=1}^n |\lambda_j| < B, \qquad \exists\, B > 0\ \forall n \geq 0.$$

  Note that the second condition is satisfied trivially if $\lambda_j > 0$, since

$$\sum_{j=1}^n \lambda_j = \int_a^b 1\; w(x)\,\mathrm{dx} = b-a.$$

- *the approximation can be improved using "Romberg integration": Say we are given a sequence $c_n \to c \in \mathbb{R}$. Then the sequence $2c_{2n} - c_n$ approximates the limit one order faster, i.e., if we have a look at the asymptotic expansion:*

$$\left.\begin{array}{ll} c_n & = c\left(1 + \frac{\gamma_1}{n} + \frac{\gamma_2}{n^2} + \frac{\gamma_3}{n^3} + \dots\right) \\ c_{2n} & = c\left(1 + \frac{\gamma_1}{2n} + \frac{\gamma_2}{4n^2} + \frac{\gamma_3}{8n^3} + \dots\right) \end{array}\right\} \Rightarrow \quad 2c_{2n} - c_n = c\left(1 - \frac{\gamma_2}{2n^2} - \frac{\gamma_3}{4n^3} - \dots\right)$$

  This procedure can be applied iteratively.

1.7 Approximation

Recall that we are always considerung proper weight functions. Then the inner product $\langle\cdot,\cdot\rangle_w$ induces a norm $\|f\| := \sqrt{\langle f,f\rangle_w}$ on the weighted L^2-space $L^2_w(I)$.

Theorem 1.17. *Suppose $f \in L^2_w(a,b)$, $(\phi_n(x))_{n\geq 0}$ a sequence of* orthonormal *polynomials with respect to w. Furthermore, let*

$$Q(x) = \sum_{k=0}^{n} a_k \phi_k(x), \qquad a_n \neq 0.$$

Then the integral

$$\int_a^b (Q(x) - f(x))^2 \, w(x) \, \mathrm{dx}$$

becomes minimal when $a_k = \int_a^b f(x)\phi_k(x)w(x)\,\mathrm{dx}$. Moreover

$$\sum_{k=0}^{n} a_k^2 \leq \int_a^b f(x)^2 w(x)\,\mathrm{dx}. \tag{1.8}$$

Proof. We start by calculating the squared L^2_w-norm:

$$\begin{aligned} 0 &\leq \int_a^b (Q(x)-f(x))^2\, w(x)\,\mathrm{dx} = \int_a^b \left(\sum_{k=0}^{n} a_k\phi_k(x) - f(x)\right)^2 w(x)\,\mathrm{dx} \\ &= \sum_{j,k=0}^{n} a_j a_k \underbrace{\int_a^b \phi_j(x)\phi_k(x)w(x)\,\mathrm{dx}}_{\delta_{jk}} - 2\sum_{k=0}^{n} a_k \int_a^b \phi_k(x)f(x)w(x)\,\mathrm{dx} + \int_a^b f(x)^2 w(x)\,\mathrm{dx} \\ &= \sum_{k=0}^{n} a_k^2 - 2\sum_{k=0}^{n} a_k \int_a^b \phi_k(x)f(x)w(x)\,\mathrm{dx} + \int_a^b f(x)^2 w(x)\,\mathrm{dx}. \end{aligned}$$

If we define the Fourier coefficients $c_k = \int_a^b \phi_k(x)f(x)w(x)\,\mathrm{dx}$ and add and subtract $\sum_{k=0} c_k^2$ then from the above it follows that

$$0 \leq \sum_{k=0}^{n} (a_k - c_k)^2 + \int_a^b f(x)^2 w(x)\,\mathrm{dx} - \sum_{k=0}^{n} c_k^2.$$

The right hand side becomes minimal, if $a_k = c_k$. With this choice we also have (1.8). □

Remark 1.18.

For $f \in L^2_w(a,b)$ the computation of a_k is well defined, since

$$\left| \int_a^b x^k f(x) w(x)\,\mathrm{dx} \right| < \infty$$

by Cauchy-Schwarz inequality (proof: exercise).

Recall the definition of kernel polynomials (now for an orthonormal sequence, i.e. $h_n = 1$) $k_n(x,y) = \sum_{j=0}^{n} \phi_j(x)\phi_j(y)$. Kernel polynomials satisify the reproducing property: Let $p \in K[x]$, $\deg p(x) \leq n$:

$$\int_a^b p(x) k_n(x,y) w(x)\,\mathrm{dx} = p(y).$$

Chapter 2

Special families of orthogonal polynomials

Next we review some of the classicial families of orthogonal polynomials and some of their characteristic properties.

2.1 Chebyshev polynomials

Definition 2.1. *(Chebyshev polynomials of the first kind) For $n \geq 0$, $x = \cos\theta$, $0 \leq \theta \leq \pi$, we define $T_n(x) = \cos(n\theta)$.*

Some properties of Chebyshev polynomials that are immediate consequences of the definition are

$$T_n(1) = 1, \quad T_n(-1) = (-1)^n, \quad |T_n(x)| \leq 1, \quad T_n(-x) = (-1)^n T_n(x),$$

for $n \geq 0$. The orthogonality relation and the three term recurrence follow from elementary results on trigonometric functions.

Theorem 2.2. *For $m, n \geq 0$ we have*

$$\int_{-1}^{1} T_n(x) T_m(x) \frac{\mathrm{dx}}{\sqrt{1-x^2}} = 0, \qquad m \neq n, \tag{2.1}$$

$$T_{n+1}(x) - 2x T_n(x) + T_{n-1}(x) = 0, \qquad n \geq 1, \quad T_0(x) = 1, \ T_1(x) = x. \tag{2.2}$$

Proof. For (2.1) substitute $x = \cos\theta$, (2.2) follows from the addition formula

$$2 \cos m\theta \cos n\theta = \cos(m+n)\theta + \cos(n-m)\theta$$

for $m = 1$. □

From the three term recurrence (2.2) and the initial values $T_0(x) = 1, T_1(x) = x$ it follows that $T_n(x)$ have *integer coefficients* and that $\mathrm{lc}(T_n(x)) = 2^{n-1}$ for $n \geq 1$. Note that the addition formula used in the proof above gives a simple *linearization formula* for products of Chebyshev polynomials:

$$T_n(x) T_m(x) = \tfrac{1}{2} \left(T_{n+m}(x) + T_{n-m}(x) \right).$$

The base case of a normal sequence of polynomials with a simple linearization formula is the monomial basis $\phi_n(x) = x^n$:

$$\phi_n(x)\phi_m(x) = x^n x^m = x^{n+m} = \phi_{n+m}(x).$$

In general for a sequence of orthonormal polynomials the linearization formula is of the form

$$\phi_n(x)\phi_m(x) = \sum_{k=0}^{n+m} a(k,m,n)\phi_k(x),$$

with $a(k,m,n) = \int_a^b \phi_n(x)\phi_m(x) \cdot \phi_k(x)w(x)\ \mathrm{dx}$. For classical orthogonal polynomials these linearization coefficients are known.

One of the most important properties of Chebyshev polynomials of the first kind was discovered by Chebyshev himself. The polynomial $2^{-n+1}T_n(x)$ is the monic polynomial of degree n that has the least deviation from zero.

Theorem 2.3. *Let $\phi_n(x)$ be a monic polynomial of degree n s.t.*

$$|\phi_n(x)| \leq 2^{-n+1} \qquad \text{for} \quad -1 \leq x \leq 1.$$

Then $\phi_n(x) = 2^{-n+1}T_n(x)$.

Proof. The only thing we need to prove this theorem is that

$$T_n\left(\cos\frac{k\phi}{n}\right) = \cos k\pi = (-1)^k, \qquad 0 \leq k \leq n.$$

Consider the polynomial $Q(x) = 2^{-n+1}T_n(x) - \phi_n(x)$. Since the leading coefficients cancel, $\deg Q(x) \leq n-1$. Since by assumption $|\phi_n(x)| \leq 2^{-n+1}$ at $x_k = \cos\frac{k\pi}{n}$ we must have $\operatorname{sign} Q(x_k) = \operatorname{sign}(T_n(x_k))$ for all $0 \leq k \leq n$. Thus at the endpoints of $[x_{k+1}, x_k]$ $Q(x)$ has different signs, so $Q(x)$ has a root in each of these segments (see the left plot in figure 2.1.

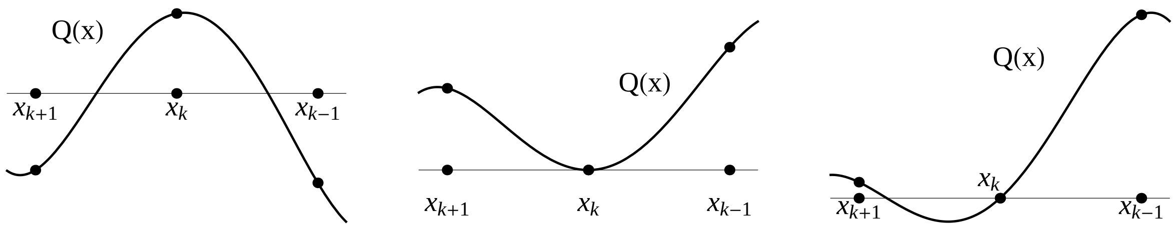


Figure 2.1: Different cases for $Q(x)$

If $Q(x_k) = 0$ then either x_k is a double root as depicted in the middle plot, or either within $[x_{k+1}, x_k]$ or within $[x_k, x_{k-1}]$ there is more than one root.

But there are exactly n segments, thus $Q(x)$ has at least n roots which implies that Q vanishes identically. □

Chebyshev polynomials of the first kind are denoted by $T_n(x)$ because of another common transcript of his name "Tchebycheff" (also other spellings exist). The other types of Chebyshev polynomials are then named in alphabetical order, we will only discuss the second kind here.

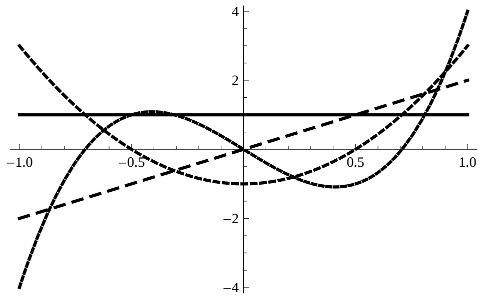


Figure 2.2: Chebyshev polynomials of the second kind $U_n(x)$ for $n = 0, 1, 2, 3$

Definition 2.4. *(Chebyshev polynomials of the second kind) For $n \geq 0$, $x = \cos\theta$ and $0 \leq \theta \leq \pi$ we define*

$$U_n(x) = \frac{\sin(n+1)\theta}{\sin\theta}.$$

Again orthogonality and the three term recurrence follow from elementary identities on trigonometric functions.

Theorem 2.5. *For $n, m \geq 0$ and $-1 \leq x \leq 1$ we have*

$$\int_{-1}^{1} U_n(x)U_m(x)\sqrt{1-x^2}\,\mathrm{dx} = 0, \qquad m \neq n, \tag{2.3}$$

$$U_{n+1}(x) - 2xU_n(x) + U_{n-1}(x) = 0, \qquad n \geq 1, \quad U_0(x) = 1,\ U_1(x) = 2x. \tag{2.4}$$

Proof. analogous to theorem 2.2. □

Note that the three term recurrence is the same as for Chebyshev polynomials of the first kind, only the initial values differ. The first few instances of Chebyshev polynomials of the second kind are given by:

$$U_0(x) = 1, \quad U_1(x) = 2x, \quad U_2(x) = 4x^2 - 1, \quad \text{and} \quad U_3(x) = 4x(2x^2 - 1).$$

Some immediate consequences of the definition are

$$\mathrm{lc}\, U_n(x) = 2^n, \quad U_n(1) = n + 1, \quad U_n(-x) = (-1)^n U_n(x),$$

for $n \geq 0$ and that $U_n(x)$ have all integer coefficients. Also several identities relating Chebyshev polynomials of the first and second kind follow from trigonometric addition formulas, like e.g.,

$$\sin(\alpha + \beta) = \sin\alpha\cos\beta + \sin\beta\cos\alpha \tag{2.5}$$

$$\cos(\alpha + \beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta \tag{2.6}$$

If we plug in $\alpha = n\theta, \beta = \theta$ in (2.5), we obtain

$$\sin(n+1)\theta = \sin n\theta\cos\theta + \sin\theta\cos n\theta \qquad |: \sin\theta$$

$$\frac{\sin(n+1)\theta}{\sin\theta} = \cos\theta\frac{\sin n\theta}{\sin\theta} + \cos n\theta.$$

With $x = \cos\theta$ this yields

$$U_n(x) = xU_{n-1}(x) + T_n(x), \tag{2.7}$$

and analogously choosing $\alpha = (n+1)\theta$ and $\beta = \theta$ in (2.6) yields

$$T_{n+2}(x) = xT_{n+1}(x) - (1-x^2)U_n(x). \tag{2.8}$$

For the derivative of $T_n(x)$ we have using $x = \cos\theta$, i.e., $\theta(x) = \arccos x$, i.e., $\theta'(x) = -\frac{1}{\sqrt{1-x^2}}$. So, all in all,

$$T_n'(x) = -\sin n\theta \cdot n\theta'(x) = n\frac{\sin n\theta}{\sin\theta},$$

or in other words

$$T_n'(x) = nU_{n-1}(x). \tag{2.9}$$

Plugging in (2.9) into (2.7) and (2.8), respectively, yields two identities relation Chebyshev polynomials of the first kind with its derivatives:

$$\begin{aligned} T_n(x) &= \frac{1}{n+1}T_{n+1}'(x) - \frac{x}{n}T_n'(x) && (2.10) \\ (1-x^2)T_{n+1}'(x) &= (n+1)\left(xT_{n+1}(x) - T_{n+2}(x)\right). && (2.11) \end{aligned}$$

Remark 2.6. *W. Hahn (1935) showed that if the derivatives of orthogonal polynomials form a set of orthogonal polynomials, then the original set were Jacobi, Hermite or Laguerre polynomials.*

2.2 Legendre polynomials

Legendre polynomials are orthogonal with respect to the constant weight function $w(x) \equiv 1$ and on the interval $[-1, 1]$. One common way to define them is using their representation using the *Rodrigues formula*.

Definition 2.7. *(Legendre polynomials) For $n \geq 0$ and $x \in [-1, 1]$ let*

$$P_n(x) := \frac{(-1)^n}{2^n n!}\frac{d^n}{dx^n}\left(1-x^2\right)^n.$$

From the definition it is obvious that $P_n(x)$ are polynomials and that $\deg P_n(x) = n$ for all $n \in \mathbb{N}$, i.e., Legendre polynomials form a normal sequence.

Theorem 2.8. *For $n, m \geq 0$ and $x \in [-1, 1]$ we have*

$$\int_{-1}^{1} P_n(x)P_m(x)\,\mathrm{dx} = 0, \qquad n \neq m, \tag{2.12}$$

$$P_n(x) = \sum_{k=0}^{n}\binom{n}{k}^2\left(\frac{x-1}{2}\right)^{n-k}\left(\frac{x+1}{2}\right)^k. \tag{2.13}$$

Proof. For proving the orthogonality relation (2.12) we assume w.l.o.g. that $n > m$. Using Rodrigues formula and partial integration we obtain

$$\begin{aligned} \frac{(-1)^n}{2^n n!}\int_{-1}^{1} D^n\left[(1-x^2)^n\right]P_m(x)\,\mathrm{dx} &= \frac{(-1)^n}{2^n n!}\left[D^{n-1}\left[(1-x^2)^n\right]P_m(x)\right]_{x=-1}^{1} \\ &\quad - \frac{(-1)^n}{2^n n!}\int_{-1}^{1} D^{n-1}\left[(1-x^2)^n\right]P_m'(x)\,\mathrm{dx}. \end{aligned}$$

In the boundary term above the derivative of $(1-x^2)^n$ vanishes for $x = \pm 1$ for $n \geq 0$. If we iterate this procedure, after k steps we have

$$\int_{-1}^{1} P_n(x) P_m(x) \,\mathrm{dx} = (-1)^k \frac{(-1)^n}{2^n n!} \int_{-1}^{1} D^{n-k} \left[(1-x^2)^n\right] P_m(x) \,\mathrm{dx}.$$

For $k = m+1$, the kth derivative of $P_m(x)$ vanishes and this yields the orthogonality for $n > m$. By symmetry, this gives (2.12) for $n \neq m$.

For the derivation of the sum representation of Legendre polynomials we start again from Rodrigues' formula and employ Leibniz' rule, i.e.,

$$D^n(fg) = \sum_{k=0}^{n} \binom{n}{k} D^k f D^{n-k} g.$$

Pulling the factor $(-1)^n$ inside the derivative, we have in the first step

$$P_n(x) = \frac{1}{2^n n!} D^n \left[(x-1)^n (x+1)^n\right] = \frac{1}{2^n} n! \sum_{k=0} \binom{n}{k} D^k \left[(x-1)^n\right] D^{n-k} \left[(x+1)^n\right].$$

Since

$$D^k \left[(x-1)^n\right] = \frac{n!}{(n-k)!} (x-1)^{n-k} \quad \text{and} \quad D^{n-k} \left[(x+1)^n\right] = \frac{n!}{k!} (x+1)^k$$

collecting terms this yields (2.13). □

Some immediate consequences from the sum representation are

$$P_n(x) = 1, \quad P_n(-1) = (-1)^n, \quad \text{and} \quad P_n(-x) = (-1)^n P_n(x),$$

for $n \geq 0$. Having a sum representation at hand, the three term recurrence for Legendre polynomials, the Legendre differential equation and mixed difference-differential relations can be found algorithmically as we will see later in this lecture. For sake of completeness we sketch how to obtain these results using classical methods.

Theorem 2.9. *For $n \geq 0$ and $x \in [-1, 1]$ we have:*

$$(2n+1) P_n(x) = P'_{n+1}(x) - P'_{n-1}(x) \tag{2.14}$$
$$(n+1) P_n(x) = P'_{n+1}(x) - x P'_n(x) \tag{2.15}$$
$$n P_n(x) = x P'_n(x) - P'_{n-1}(x) \tag{2.16}$$
$$(1-x^2) P'_n(x) = n \left[P_{n-1}(x) - x P_n(x)\right] \tag{2.17}$$

Furthermore, Legendre polynomials satisfy the three term recurrence

$$(n+1) P_{n+1}(x) - (2n+1) x P_n(x) + n P_{n-1}(x) = 0, \tag{2.18}$$

with $P_{-1}(x) = 0$ and $P_0(x) = 1$, and they are a solution to the Legendre differential equation

$$(1-x^2) y''(x) - 2x y'(x) + n(n+1) y(x) = 0. \tag{2.19}$$

Proof. (Sketch)
ad (2.14): Write the polynomials on the right hand side using Rodrigues' formula:

$$\begin{aligned} P'_{n-1}(x) &= \frac{(-1)^{n-1}}{2^{n-1}(n-1)!} D^n \left[(1-x^2)^{n-1}\right], \\ P'_{n+1}(x) &= \frac{(-1)^{n+1}}{2^{n+1}(n+1)!} D^n \left[D^2 \left[(1-x^2)^{n+1}\right]\right]. \end{aligned}$$

Carry out the inner differentiation and simplify $P'_{n-1}(x) - P'_{n+1}(x)$.

ad (2.15): By Leibniz' rule we have in operator notation that $xD = Dx - 1$ and a repeated application of this rule yields $xD^m = D^m x - mD^{m-1}$. Apply this rule to the identity (Exercise!)

$$P'_{n+1}(x) - xP'_n(x) = \frac{(-1)^n}{2^n n!} \left[D^{n+1}\left[x(1-x^2)^n\right] + xD^{n+1}\left[(1-x^2)^n\right]\right].$$

Identity (2.16) is just the difference of (2.14) and (2.15), and (2.17) follows from (2.15) and (2.16).

To obtain the recurrence relation eliminate the derivatives using (2.16) and (2.17) and for the differential equation, eliminate the differences using (2.15) and (2.16). □

In remark 1.9 we derived an explicit formula for the weighted, squared L^2-norm h_n given the three term recurrence:

$$h_n = (-1)^n \frac{a_0}{a_n} c_n c_{n-1} \dots c_1 h_0.$$

By theorem 2.9, equation(2.18) we have that $a_n = \frac{2n+1}{n+1}$, $c_n = -\frac{n}{n+1}$. Hence with

$$h_n = (-1)^n \frac{2(n+1)}{2n+1} \left(-\frac{n}{n+1}\right)\left(-\frac{n-1}{n}\right) \cdots \left(-\frac{1}{2}\right) h_0$$

and $h_0 = 2$ all in all we obtain that

$$\int_{-1}^{1} P_n(x) P_m(x)\, \mathrm{dx} = \frac{2}{2n+1}\delta_{nm}, \qquad n, m \geq 0.$$

Another interesting property of Legendre polynomials can be derived from the Legendre differential equation.

Theorem 2.10. *Let $n \geq 2$. The successive relative maxima of $|P_n(x)|$, when x increases from 0 to 1 form an increasing sequence.*

Proof. For the proof we will consider $P_n(x)^2$ instead of the absolute value since this function shares the same properties but has the advantage of being differentiable. In the first step we define an envelope $f_n(x)$ for $P_n(x)^2$ by

$$n(n+1) f_n(x) = n(n+1) P_n(x)^2 + (1-x^2) P'_n(x)^2.$$

Since all expressions on the right hand side are non-negative $f_n(x)$ clearly is an upper bound for $P_n(x)^2$. Furthermore for critical points z with $P'_n(z)$ we have that $f_n(z) = P_n(z)$ and also on the endpoints of the interval $f_n(\pm 1) = P_n(\pm 1)^2$.

In the second step we compute the first derivative of the envelope:

$$\begin{aligned} n(n+1)f_n'(x) &= 2n(n+1)P_n(x)P_n'(x) - 2xP_n'(x)^2 + 2(1-x^2)P_n'(x)P_n''(x) \\ &= 2P_n'(x)\left[n(n+1)P_n(x) - xP_n'(x) + (1-x^2)P_n''(x)\right]. \end{aligned}$$

Now we recall the Legendre differential equation (2.19)

$$(1-x^2)y''(x) - 2xy'(x) + n(n+1)y(x) = 0.$$

Hence $n(n+1)f_n'(x) = 2xP_n'(x)^2$ which is greater or equal to zero if $0 \le x \le 1$. □

One immediate consequence of this result is that Legendre polynomials are in absolute value bounded by one, i.e., $|P_n(x)| \le |P_n(x)| \le 1$.

2.3 Jacobi polynomials

Jacobi polynomials are orthogonal with respect to the weight function $w(x) = (1-x)^\alpha(1+x)^\beta$ for $\alpha, \beta > -1$ on the interval $[-1,1]$. We will also state their definition in terms of the Rodrigues formula.

Definition 2.11. *(Jacobi polynomials) For $\alpha, \beta > -1$, $-1 \le x \le 1$ and $n \ge 0$ define*

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

From the definition it is obvious that $\deg P_n^{(\alpha,\beta)}(x) = n$ for all $n \in \mathbb{N}$.

Theorem 2.12. *For $n, m \ge 0$:*

$$\int_{-1}^{1}(1-x)^\alpha(1+x)^\beta P_n^{(\alpha,\beta)}(x)P_m^{(\alpha,\beta)}(x)\,\mathrm{dx} = 0, \qquad n \ne m.$$

Proof. Completely analogous to the proof of the orthogonality of Legendre polynomials, see theorem 2.13 (Exercise). □

We have already encountered some special cases of Jacobi polynomials:

- Legendre polynomials: $\alpha = \beta = 0,\ w(x) = 1,\ P_n(x) = P_n^{(0,0)}(x)$
- Chebyshev polynomials of the first kind: $\alpha = \beta = -\frac{1}{2}$, $T_n \simeq P_n^{(-\frac{1}{2},-\frac{1}{2})}(x)$ (equal up to normalization)
- Chebyshev polynomials of the second kind: $\alpha = \beta = \frac{1}{2}$, $U_n \simeq P_n^{(\frac{1}{2},\frac{1}{2})}(x)$

Jacobi polynomials for $\alpha = \beta$ are also called *Gegenbauer polynomials* or *ultraspherical polynomials* and are usually introduced using a different normalization. The relation to Jacobi polynomials is given by

$$\alpha = \beta = \lambda - \tfrac{1}{2} \quad \text{for} \quad \lambda > -\tfrac{1}{2}: \quad C_n^\lambda(x) = \frac{(2\lambda)_n}{\left(\lambda + \frac{1}{2}\right)_n} P_n^{(\lambda-\frac{1}{2},\lambda-\frac{1}{2})}(x).$$

Starting from the Rodrigues formula using Leibniz rule also for Jacobi polynomials a sum representation can be derived.

Theorem 2.13. *For $n \geq 0$, $\alpha, \beta > -1$ and $-1 \leq x \leq 1$:*

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

Another commonly used sum representation for Jacobi polynomials is given by

$$P_n^{(\alpha,\beta)}(x) = \frac{(\alpha+1)_n}{n!} \sum_{k\geq 0} \frac{(-n)_k (n+\alpha+\beta+1)_k}{(\alpha+1)_k k!} \left(\frac{1-x}{2}\right)^{k}. \tag{2.20}$$

Note, that the sum above is finite because of the factor $(-n)_k$ in the numerator. Some immediate consequences of these sum representations are

$$P_n^{(\alpha,\beta)}(1) = \frac{(\alpha+1)_n}{n!}, \quad P_n^{(\alpha,\beta)}(-x) = (-1)^n P_n^{(\beta,\alpha)}(x), \quad \text{and} \quad P_n^{(\alpha,\beta)}(1) = (-1)^n \frac{(\beta+1)_n}{n!}.$$

Derivatives of Jacobi polynomials are again Jacobi polynomials with shifted parameters which is readily calculated. For $n \geq 0$ we have using the sum representation (2.20)

$$\begin{aligned} \frac{d}{dx} P_n^{(\alpha,\beta)}(x) &= \frac{(\alpha+1)_n}{n!} \sum_{k\geq 0} \frac{(-n)_k (n+\alpha+\beta+1)_k}{(\alpha+1)_k k!} k \left(\frac{1-x}{2}\right)^{k-1} \left(-\frac{1}{2}\right) \\ &= -\frac{1}{2} \frac{(\alpha+1)_n}{n!} \sum_{k\geq 1} \frac{(-n)_k (n+\alpha+\beta+1)_k}{(\alpha+1)_k (k-1)!} \left(\frac{1-x}{2}\right)^{k-1}. \end{aligned}$$

Rewriting the Pochhammer symbols as $(a)_{n+1} = a(a+1)_n$ the above equals to

$$-\frac{1}{2} \frac{(\alpha+1)_n}{n!} \frac{(-n)(n+\alpha+\beta+1)}{\alpha+1} \sum_{k\geq 1} \frac{(-n+1)_{k-1} (n+\alpha+\beta+2)_{k-1}}{(\alpha+2)_{k-1} (k-1)!} \left(\frac{1-x}{2}\right)^{k-1}.$$

Cancelling and identifying the parameters on the right hand side we have thus obtained

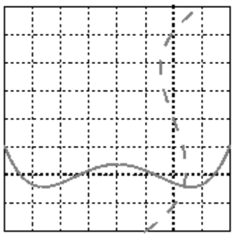
Theorem 2.14. *For $n \geq 0$:*

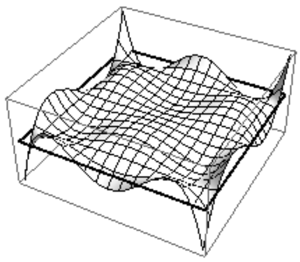
$$\frac{d}{dx} P_n^{(\alpha,\beta)}(x) = \frac{n+\alpha+\beta+1}{2} P_{n-1}^{(\alpha+1,\beta+1)}(x).$$

Since Chebyshev polynomials can be written as Jacobi polynomials identity (2.9) is just a special case of theorem 2.14. For Legendre polynomials it follows that that

$$P_n'(x) = \frac{n+1}{2} P_{n-1}^{(1,1)}(x).$$

Note that all identities in theorem 2.9 have generalizations to identities between different families of Jacobi polynomials. Identities of this type can be found in many text books, in the handbook of mathematical functions [1] or in online libraries such as the digital library of mathematical functions (DLMF) or the Wolfram functions site. Later we will see how to find or prove this type of relations in an algorithmic manner.





So far we defined orthogonal polynomials on real intervals only. In applications such as, e.g., finite element methods, polynomial bases defined on simple geometric objects such as squares or triangles are used. An orthogonal basis for the space of bivariate polynomials defined on the unit square $Q = [-1,1]^2$ that is orthogonalwith respect to the inner product

$$\langle f, g\rangle_{L^2 Q} = \int_Q f(x,y)g(x,y)\;\mathrm{d}(x,y)$$

is given by the following tensor product construction using Legendre polynomials:

$$\phi_{i,j}(x,y) = P_i(x)P_j(y), \qquad i,j \geq 0,$$

the plot illustrates the case $i = 4, j = 3$. Orthogonality follows from the orthogonality of the components on $[-1,1]$, since

$$\int_Q \phi_{i,j}(x,y)\phi_{k,l}(x,y)\;\mathrm{d}(x,y) = \int_{-1}^{1} P_i(x)P_k(x)\;\mathrm{dx}\int_{-1}^{1} P_j(y)P_l(y)\;\mathrm{dy} = \frac{2\delta_{ik}}{2i+1}\frac{2\delta_{jl}}{2j+1}.$$

If we want to construct an orthogonal basis on a triangle then we cannot directly proceed by a tensor product construction, the following approach based on Dubiner [6] uses a tensor product-like construction and uses properly chosen Jacobi polynomials.

We will consider the triangle $T = \left\{(x,y) \mid -1 \leq y \leq 1, -\frac{1-y}{2} \leq x \leq \frac{1-y}{2}\right\}$ (i.e., the triangle with vertices $(-1,-1),(1,-1)$ and $(0,1)$) and view it as a collapsed version of the square Q. Then the family of polynomials orthogonal with respect to the inner product

$$\langle f, g\rangle_{L^2(T)} = \int_T f(x,y)g(x,y)\;\mathrm{d}(x,y)$$

is given by

$$\phi_{i,j}(x,y) = P_i\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i P_j^{(2i+1,0)}(y), \quad i,j \geq 0.$$

First of all note that the functions defined above are indeed polynomials because of the compensating factor $(1-y)^i$. In order to show orthogonality we compute the integral by decoupling the integrands using a substitution that is commonly known as Duffy transformation:

$$z = \frac{2x}{1-y} \quad \Rightarrow \quad \mathrm{dz} = \frac{2\;\mathrm{dx}}{1-y}.$$

First notice that the integration over the triangle amounts to

$$\int_T \phi_{i,j}(x,y)\phi_{k,l}(x,y)\;\mathrm{d}(x,y) = \int_{-1}^{1}\int_{-\frac{1-y}{2}}^{\frac{1-y}{2}} \phi_{i,j}(x,y)\phi_{k,l}(x,y)\;\mathrm{dx}\;\mathrm{dy}$$

By means of the Duffy substitution the integrals decouple and we obtain further

$$\int_{-1}^{1} P_i(z)P_k(z)\;\mathrm{dz}\int_{-1}^{1}\left(\frac{1-y}{2}\right)^{i+k+1} P_j^{(2i+1,0)}(y)P_l^{(2k+1,0)}(y)\;\mathrm{dy}.$$

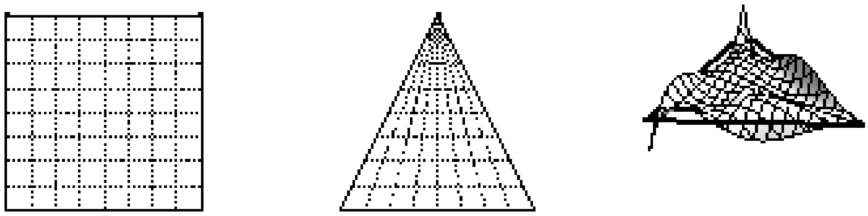


Figure 2.3: Square collapsing to triangle with singular vertex marked and $\phi_{3,2}(x,y)$

Evaluating the integrals - first the integration with respect to z and then plugging in the Kronecker delta and integrating with respect to y - we end up with

$$\begin{aligned}\int_T \phi_{i,j}(x,y)\phi_{k,l}(x,y)\ \mathrm{d}(x,y) &= \frac{2}{2i+1}\int_{-1}^{1}\left(\frac{1-y}{2}\right)^{2i+1} P_j^{(2i+1,0)}(y)P_l^{(2i+1,0)}(y)\ \mathrm{dy}\\ &= \frac{2}{2i+1}\frac{1}{i+j+1}\delta_{ik}\delta_{jl}.\end{aligned}$$

2.4 Hermite and Laguerre polynomials

Both families of orthogonal polynomials introduced next are closely related to probability distributions. They differ from the previously discussed polynomials as they are defined over infinite intervals.

2.4.1 Hermite polynomials

The function $f(x) = \mathrm{e}^{-x^2}$ defined over $\mathbb{R} = (-\infty, \infty)$ has many interesting properties such as it essentially equals its own Fourier transform, i.e.,

$$\hat{f}(x) = \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty} f(t)\mathrm{e}^{2\mathrm{i}xt}\ \mathrm{dt} = f(x). \tag{2.21}$$

Since the integral is well defined this is easy to be seen by first noting that

$$\mathrm{e}^{2\mathrm{i}xt} = \cos(2xt) + \mathrm{i}\sin(2xt).$$

The function $f(x)$ is even, so is cosine, and sine is an odd function. Thus the integral over the product $f(x)\sin(2xt)$ vanishes and exploiting the symmetry of $f(t)\cos(2xt)$ we thus have

$$\hat{f}(x) = \frac{2}{\sqrt{\pi}}\int_0^{\infty} f(t)\cos(2xt)\ \mathrm{dt}.$$

Differentiating the above identity with respect to x and integrating partially the differential equation

$$\frac{d}{dx}\hat{f}(x) = -2x\hat{f}(x)$$

for $\hat{f}(x)$ can be determined. This together with the inital value $\hat{f}(0) = \sqrt{\pi}$ yields (2.21). Furthermore the integral is uniformly convergent in any disk $|x| < r$ and can be bounded from above by

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \mathrm{e}^{-t^2} \mathrm{e}^{2rt} \, \mathrm{dt}$$

on the disk. Thus the integral can be differentiated with respect to x and we have

$$\frac{d}{dx^n} \mathrm{e}^{-x^2} = \frac{(2\mathrm{i})^n}{\sqrt{\pi}} \int_{-\infty}^{\infty} \mathrm{e}^{-t^2} t^n \mathrm{e}^{2\mathrm{i}xt} \, \mathrm{dt}. \tag{2.22}$$

The polynomials that are orthogonal with respect to the normal distribution are the Hermite polynomials and we state their definition again in terms of a Rodrigues type formula.

Definition 2.15. *(Hermite polynomials) For $n \geq 0$ define*

$$\mathrm{e}^{-x^2} H_n(x) = (-1)^n \frac{d^n}{dx^n} \mathrm{e}^{-x^2}.$$

From the definition it is immediate that $H_n(x)$ is a polynomial of degree n for any $n \geq 0$. The first few instances are given by

$$H_0(x) = 1, \quad H_1(x) = 2x, \quad H_2(x) = 2\left(2x^2 - 1\right), \quad H_3(x) = 4x\left(2x^2 - 3\right).$$

Together with the derivative of the Fourier integral (2.22) the definition immediately gives the integral representation

$$H_n(x) = \frac{(-2\mathrm{i})^n}{\sqrt{\pi}} \mathrm{e}^{x^2} \int_{-\infty}^{\infty} \mathrm{e}^{-t^2} t^n \mathrm{e}^{2ixt} \, \mathrm{dt} \tag{2.23}$$

for Hermite polynomials.Orthogonality can be shown again starting from the Rodrigues formula using partial integration.

Theorem 2.16. *For $n, m \geq 0$ with $n \neq m$ we have*

$$\int_{-\infty}^{\infty} H_n(x) H_m(x) \mathrm{e}^{-x^2} \, \mathrm{dx} = 0.$$

By the general theory we know that Hermite polynomials thus also satisfy a three term recurrence. In order to derive this recurrence relation we use the concept of *generating functions*. In general, given a sequence $(a_n)_{n \geq 0}$ we use as a different form of representation its generating function $F(z)$ defined formally as

$$F(z) = \sum_{n \geq 0} a_n z^n.$$

By formally we mean that this is a pure encoding of the given sequence and is not considered as a functional object in general. In particular this means that we do not require (or expect) convergence. However, if feasible, we will use common functional notation to formally express the given sequence. As the most simple example consider the constant sequence $a_n = 1$ for $n \geq 0$. It is widely known that

$$F(z) = \sum_{n \geq 0} z^n = \frac{1}{1 - z},$$

and, in the sense described above, we will consider $\frac{1}{1-z}$ a representation of the sequence $(a_n)_{n\geq 0}$. Another simple example is the generating function of $a_n = 1/n!$,

$$F(z) = \sum_{n\geq 0} \frac{z^n}{n!} = \mathrm{e}^z.$$

These objects are well defined in the *ring of formal power series*. Given a field K it is usually denoted by $K[\![x]\!]$ and becomes a ring, e.g., with $'+'$ defined as termwise addition, i.e.,

$$\sum_{n\geq 0} a_n z^n + \sum_{n\geq 0} b_n z^n = \sum_{n\geq 0} (a_n + b_n) z^n,$$

and $'\cdot'$ can defined, e.g., by the Cauchy product

$$\left(\sum_{n\geq 0} a_n z^n\right)\left(\sum_{n\geq 0} b_n z^n\right) = \sum_{n\geq 0}\left(\sum_{k=0}^{n} a_k b_{n-k}\right) z^n.$$

It is easily verified that these operations turn $K[\![x]\!]$ into a ring. Note that also the generating function of the sequence $\{a_n = n!\}_{n\geq 0}$ is an element of this ring even if there is *no* analytic function corresponding to the formal power series. Further details can be found, e.g., in [11]. Generating functions allow to operate with a given sequence on two levels, the sequence itself or its functional represenation, to obtain new identities. This is what we will do now with the sequence of Hermite polynomials.

The integral representation of Hermite polynomials (2.23) together with the formula for the Fourier transform (2.21) gives a closed form representation for the generating function of $\frac{1}{n!}H_n(x)$:

$$\begin{aligned} F(z) = \sum_{n=0} \frac{H_n(x)}{n!} z^n &= \frac{1}{\sqrt{\pi}} \mathrm{e}^{x^2} \int_{-\infty}^{\infty} \sum_{n\geq 0} \frac{(-2\mathrm{i}tz)^n}{n!} \mathrm{e}^{-t^2} \mathrm{e}^{2ixt}\,\mathrm{d}t \\ &= \frac{1}{\sqrt{\pi}} \mathrm{e}^{x^2} \int_{-\infty}^{\infty} \mathrm{e}^{-t^2} \mathrm{e}^{2it(x-z)}\,\mathrm{d}t = \mathrm{e}^{2xz-z^2}. \end{aligned} \tag{2.24}$$

We can use this closed form to derive a sum representation of Hermite polynomials.

Theorem 2.17. *For $n \geq 0$ and $x \in \mathbb{R}$*

$$H_n(x) = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n!}{k!(n-2k)!} (-1)^k (2x)^{n-2k}.$$

Proof. Starting from the closed form representation of the generating function $F(z)$ derived in (2.24) the sum representation follows by carrying out the Cauchy product $\mathrm{e}^{2xz}\mathrm{e}^{-z^2}$ and comparing coefficients:

$$\begin{aligned} \mathrm{e}^{2xz}\mathrm{e}^{-z^2} &= \sum_{n\geq 0} \frac{(2x)^n}{n!} z^n \sum_{n\geq 0} \frac{(-1)^n}{n!} z^n \\ &= \sum_{n\geq 0} \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k}{k!} \frac{(2x)^{n-2k}}{(n-k)!} z^n. \end{aligned}$$

□

Some immediate consequences of the sum representation are

$$H_n(-x) = (-1)^n H_n(x), \quad H_{2n+1}(0) = 0, \quad H_{2n}(0) = (-1)^n.$$

Further identities on Hermite polynomials can be obtained by (formally) differentiating the generating function $F(z)$. It is easily verified that

$$F'(z) = (2x - 2z)F(z).$$

On the series level, the left hand side reads as

$$F'(z) = \sum_{n\geq 1} \frac{H_n(x)}{(n-1)!} z^{n-1} = \sum_{n\geq 0} \frac{H_{n+1}(x)}{n!} z^n,$$

and for the right hand side we have

$$(2x - 2z)F(z) = \sum_{n\geq 0} \frac{2xH_n(x)}{n!} z^n - \sum_{n\geq 1} \frac{2nH_{n-1}(x)}{n!} z^n.$$

Comparing coefficients on both sides yields for $n \geq 1$ the three term recurrence for Hermite polynomials,

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x),$$

with initial values $H_0(x) = 1$, $H_1(x) = 2x$. This recurrence can be extended to $n \geq 0$ with $H_{-1}(x) = 0$. The weighted L^2-norm can again be derived using the formula from remark 1.9:

$$h_n = (-1)^n \frac{a_0}{a_n} c_n \cdot \dots \cdot c_1 h_0 = 2^n n! \sqrt{\pi},$$

since $h_0 = \int_{-\infty}^{\infty} \mathrm{e}^{-x^2} \, \mathrm{d}x = \sqrt{\pi}$. If we take the derivative of $F(z)$ with respect to x, then we obtain

$$\frac{d}{dx} F(z) = \sum_{n\geq 0} \frac{H_n'(x)}{n!} z^n,$$

and on the closed form side by shifting the summation index

$$\frac{d}{dx} \mathrm{e}^{2xz - z^2} = 2z\mathrm{e}^{2xz-z^2} = \sum_{n\geq 0} \frac{2H_n(x)}{n!} z^{n+1} = \sum_{n\geq 1} \frac{2nH_n(x)}{n!} z^n.$$

By coefficient comparison it follows that derivatives of Hermite polynomials are again Hermite polynomials, i.e.,

$$H_n'(x) = 2nH_{n-1}(x), \quad n \geq 1, \tag{2.25}$$

which can be extended to $n \geq 0$ using $H_{-1}(x) = 0$.

2.4.2 Laguerre polynomials

We conclude the introduction of the classical families of orthogonal polynomials with Laguerre polynomials $L_n^\alpha(x)$ that are defined on $[0, \infty)$ and orthogonal with respect to the gamma distribution, i.e., with respect to the weight function $w_\alpha(x) = x^\alpha \mathrm{e}^{-x}$ for $\alpha > -1$.

Definition 2.18. *(Laguerre polynomials) For $\alpha > -1$, $x \geq 0$, $n \geq 0$, let*

$$x^\alpha \mathrm{e}^{-x} L_n^\alpha(x) = \frac{1}{n!} \frac{d}{dx^n} \left[x^{n+\alpha} \mathrm{e}^{-x} \right].$$

From the definition it is obvious that $L_n^\alpha(x)$ is a polynomial of degree n for each $n \geq 0$. The first few instances are given by

$$L_0^\alpha(x) = 1, \quad L_1^\alpha(x) = 1-x, \quad L_2^\alpha(x) = \tfrac{1}{2}\left(x^2 - 4x + 2\right), \quad L_3^\alpha(x) = \tfrac{1}{6}\left(-x^3 + 9x^2 - 18x + 6\right).$$

Starting from Rodrigues formula (definition 2.18) the orthogonality relation and a sum representation for Laguerre polynomials are easily derived.

Theorem 2.19. *For $n, m \geq 0$, $\alpha > -1$ and $x \geq 0$ we have*

$$\int_0^\infty L_n^\alpha(x) L_m^\alpha(x) x^\alpha \mathrm{e}^{-x} \,\mathrm{dx} = 0, \qquad \textit{for } n \neq m, \quad L_n^\alpha(x) = \frac{(\alpha+1)_n}{n!} \sum_{k=0}^n \frac{(-n)_k}{(\alpha+1)_k k!} x^k.$$

Proof. The orthogonality relation follows by partial integration and the sum representation using Leibniz formula. □

Most common is the special case $\alpha = 0$, i.e., the polynomials orthogonal with respect to $w(x) = \mathrm{e}^{-x}$ and they are usually simply denoted by

$$L_n(x) := L_n^0(x) = \sum_{k=0}^n \frac{(-n)_k}{(k!)^2} x^k.$$

Starting from the sum representation it is more or less straight forward to derive a closed form for the generating function of Laguerre polynomials:

$$F(x; z) = \sum_{n \geq 0} L_n^\alpha(x) z^n = \frac{1}{(1-z)^{\alpha+1}} \exp\left(-\frac{xz}{1-z} \right). \tag{2.26}$$

In the first step we plug in the sum representation and use that $(-n)_k = (-1)^k n^{\underline{k}} = (-1)^k \frac{n!}{(n-k)!}$ and obtain

$$F(x; z) = \sum_{n=0}^\infty \frac{(\alpha+1)_n}{n!} \sum_{k=0}^n \frac{n!}{(n-k)!} \frac{1}{(\alpha+1)_k k!} (-x)^k z^n.$$

Then we can exchange the order of summation and shift the summation index of the inner sum

$$F(x; z) = \sum_{k=0}^\infty \frac{(-x)^k}{(\alpha+1)_k k!} \sum_{n=k}^\infty \frac{(\alpha+1)_n}{(n-k)!} z^n = \sum_{k=0}^\infty \frac{(-xz)^k}{(\alpha+1)_k k!} \sum_{n=0}^\infty \frac{(\alpha+1)_{n+k}}{k!} z^n$$

The ratio of Pochhammer symbols $\frac{(\alpha+1)_{n+k}}{(\alpha+1)_k}$ can be rewritten as $(\alpha+k+1)_n$ and we can use the (generalized) binomial theorem (proof is left as an exercise)

$$\sum_{n \geq 0} \frac{(a)_n}{n!} z^n = (1-z)^{-a}.$$

Together with the series expansion for the exponential function this concludes the proof:

$$F(x;z) = \sum_{k=0}^{\infty} \frac{(-xz)^k}{k!} \sum_{n=0}^{\infty} \frac{(\alpha+k+1)_n}{n!} z^n = (1-z)^{-\alpha-1} \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{xz}{1-z}\right)^k.$$

The closed form representation of the generating function gives us a way to derive the three term recurrence of Laguerre polynomials and a mixed difference-differential relation analogously to the derivation for Hermite polynomials.

Theorem 2.20. *For $n \geq 1$, $x \geq 0$ and $\alpha > -1$ we have*

$$(n+1)L_{n+1}^{\alpha}(x) + (x - 2n - \alpha - 1)\, L_n^{\alpha}(x) + (n+\alpha)L_{n-1}^{\alpha}(x) = 0,$$

with $L_0^{\alpha}(x) = 1$, $L_1^{\alpha}(x) = 1 - x$, and furthermore

$$\frac{d}{dx}\left(L_{n-1}^{\alpha}(x) - L_n\alpha(x)\right) = L_{n-1}^{\alpha}(x).$$

Proof. To prove the recurrence relation we first observe that the generating function $F(x;z)$ satisfies the differential equation

$$(1-z)^2 \frac{d}{dz} F(x;z) = (-x + (\alpha+1)(1-z))\, F(x;z).$$

For the left hand side we obtain by termwise differentiation and shifting summation indices that

$$(1-z)^2 \frac{d}{dz} F(x;z) = \sum_{n\geq 0}(n+1)L_{n+1}^{\alpha}(x) z^n - \sum_{n\geq 0} 2n L_n^{\alpha}(x) z^n + \sum_{n\geq 0} n L_n^{\alpha}(x) z^{n+1}.$$

Similarily we derive for the right hand side

$$(-x + (\alpha+1)(1-z))\, F(x;z) = \sum_{n\geq 0}(-x + (\alpha+1))\, L_n^{\alpha}(x) z^n - \sum_{n\geq 0}(\alpha+1) L_n^{\alpha}(x) z^{n+1}.$$

Comparing coefficients for $n \geq 1$ on both sides yields the three term recurrence.

For the derivation of the mixed relation we differentiate the generating function with respect to x and obtain this way that

$$(z-1)\frac{\partial}{\partial x} F(x;z) = zF(x;z).$$

Plugging in and shifting summation indices accordingly gives the result. □

It is possible to express Hermite polynomials in terms of Laguerre polynomials. In particular, we have for $m \geq 0$ that

$$H_{2m}(x) = (-1)^m 2^{2m} m! L_m^{-1/2}(x^2), \tag{2.27}$$

and

$$H_{2m+1}(x) = (-1)^m 2^{2m+1} m! x L_m^{1/2}(x^2). \tag{2.28}$$

We will only show the even case here, i.e., we show that for some constant $C = C(m)$ we have that $H_{2m}(x) = C(m)L_m^{-1/2}(x^2)$. For this it is sufficient to show that for any polynomial $q(x)$ with degree less than $2m$ the orthogonality condition

$$\int_{-\infty}^{\infty} L_m^{-1/2}(x^2)q(x)\mathrm{e}^{-x^2}\,\mathrm{dx} = 0 \tag{2.29}$$

holds. Laguerre polynomials $L_m^\alpha(x^2)$ are obviously even on $\mathbb{R}$ and so the integral vanishes if q is odd. However, any polynomial $q(x)$ can be written as the sum of its odd and even part, i.e., it can be split into

$$q_{odd}(x) = \tfrac{1}{2}\left(q(x) - q(-x)\right), \quad \text{and} \quad q_{even}(x) = \tfrac{1}{2}\left(q(x) + q(-x)\right).$$

Hence we can restrict ourselves to even polynomials which, in turn, can be written as $q(x) = r(x^2)$ for some polynomial r of degree less than m. If we substitute $s = x^2$ in the integral (2.29) this yields

$$\int_{-\infty}^{\infty} L_m^{-1/2}(x^2)q(x)\mathrm{e}^{-x^2}\,\mathrm{dx} = \frac{1}{2}\int_0^{\infty} L_m^{-1/2}(s)r(s)e^{-s}s^{-1/2}\,\mathrm{ds},$$

which vanishes by the orthogonality of Laguerre polynomials.

Chapter 3

Symbolic computation

Orthogonal polynomials belong to the class of holonomic functions - a class for which over the past decades several algorithms were developed that are capable of solving problems as finding closed forms for sums or integrals. What we understood as "closed form" depends on the context - it may be an explicit, short form, it may be a description in terms of a defining difference or differential equation. For further reading we refer, e.g., to the textbooks of Petkovšek, Wilf and Zeilberger [14], Kauers and Paule [11] or Wilf [22].

3.1 The holonomic universe

The ring of formal power series that we introduced in the section on Hermite polynomials is part of the holonomic universe. We can choose between defining holonomicity from the generating functions point of view or from the sequence level. We do the latter.

Definition 3.1. *A sequence $(a_n)_{n\geq 0}$ over a field K is called* holonomic *iff there exist polynomials $q_0, \dots, q_d \in K[x]$, $q_d \neq 0$, such that for all $n \in \mathbb{N}$*

$$q_0(n)a_n + q_1(n)a_{n+1} + \cdots + q_d(n)a_{n+d} = 0.$$

If all the q_i are constant (i.e., from the ground field K), then we refer to the sequence as C-finite*, otherwise also as* P-finite*.*

The integer d in the definition above is called the *order* of the recurrence. A particular instance of holonomic sequences are sequences that satisfy recurrence relations of order 1, i.e., sequences whose shift quotient is a rational function.

Definition 3.2. *A sequence $(a_n)_{n\geq 0}$ with elements in K is called* hypergeometric *over K, if there exist polynomials $p, q \in K[x]$ such that the linear relation*

$$p(n)a_{n+1} + q(n)a_n = 0$$

is satisfied for all $n \geq 0$. A bivariate sequence $(a(n,k))_{n,k\geq 0}$ is called hypergeometric if it is hypergeometric in both variables n and k.

A hypergeometric term $a(n,k)$ is called proper hypergeometric over K if the polynomials p_i, $q_i \in K[x,y]$ satisfying

$$p_1(n,k)a(n+1,k) + p_0(n,k)a(n,k) = 0, \qquad q_1(n,k)a(n,k+1) + q_0(n,k)a(n,k) = 0,$$

split into integer linear factors of the form $\alpha n + \beta k + \gamma$, α, $\beta \in \mathbb{Z}$, $\gamma \in K$.

We have encountered hypergeometric sequences as coefficients in the sum representation of orthogonal polynomials. In theorem 2.8 we showed that Legendre polynomials can be expanded as

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

The shift quotient of $c_{n,k}(x)$ in the summation variable k is given by

$$\frac{c_{n,k+1}(x)}{c_{n,k}(x)} = \frac{(n-k)^2}{(k+1)^2}\frac{x+1}{x-1},$$

which is a rational function over the field $K(n, x)$. It is even proper hypergeometric in n and k since also

$$\frac{c_{n+1,k}(x)}{c_{n,k}(x)} = \frac{(n+1)^2(x-1)}{2(n-k+1)^2}.$$

The holonomicity of elements of the ring of formal power series is defined via holonomicity of the coefficient sequence.

Definition 3.3. *A formal power series $f(x) = \sum_{n\geq 0} a_n x^n \in K[\![x]\!]$ is called holonomic, iff $(a_n)_{n\geq 0}$ is a holonomic sequence.*

Holonomic sequences were defined as solutions to linear difference equations with polynomial coefficients. Analogously, holonomic functions can be characterized as solutions to linear differential equations with polynomial coefficients.

Lemma 3.4. *A formal power series $f(x) = \sum_{n\geq 0} a_n x^n$ is holonomic if and only if there exist polynomials $p_0, \ldots, p_d \in K[x]$, $p_d \neq 0$, such that*

$$p_0(x)f(x) + p_1(x)f'(x) + \cdots + p_d(x)f^{(d)}(x) = 0.$$

Proof. The link for transferring the difference equation satisfied by a_n to a differential equation satisfied by $f(x)$ (and vice versa) is $D_x^k x^n = n^{\underline{k}} x^{n-k}$. If the coefficients $q_i(n)$ are expanded in the basis of falling factorials, then the conversion boils down to a mere rewriting. The maximal degree of the coefficients in the difference equation translates into the order of the differential equation. Details are left as an exercise. □

Already for Hermite polynomials we have used this equivalence to derive the three term recurrence

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$$

starting from the differential equation of the generating function $F(z) = \sum_{n\geq 0} H_n(x)\frac{z^n}{n!}$. As an example we compute the closed form for the generating function of Chebyshev polynomials of the first kind starting from their defining difference equation.

Example 3.5. *Let $F(z) = \sum_{n\geq 0} T_n(x)z^n$, where $T_n(x)$ denote the Chebyshev polynomials of the first kind defined by*

$$T_{n+2}(x) - 2xT_{n+1}(x) + T_n(x) = 0, \quad T_0(x) = 1,\ T_1(x) = x.$$

In order to derive the closed form for $F(z)$ we first multiply the equation above by z^{n+2} and then we sum over all $n \in \mathbb{N}$:

$$\sum_{n\geq 0} T_{n+2}(x)z^{n+2} - 2x \sum_{n\geq 0} T_{n+1}(x)z^{n+2} + \sum_{n\geq 0} T_n(x)z^{n+2} = 0.$$

By shifting summation indices and using the definition of $F(z)$, the above can be rewritten as

$$(F(z) - xz - 1) - 2xz\,(F(z) - 1) + z^2 F(z) = 0.$$

The initial values of Chebyshev polynomials enter in the compensating terms above. Solving this (differential) equation yields the closed form solution

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

Note that for any C-finite sequence the generating function is a rational function. Holonomic objects are closed under certain operations, some of which are summarized in the following theorem.

Theorem 3.6. *Let $(a_n)_{n\geq 0}$ and $(b_n)_{n\geq 0}$ be holonomic sequences. Then the sequence $(c_n)_{n\geq 0}$ defined by*

$$c_n = a_n + b_n \tag{3.1}$$

$$c_n = a_n \cdot b_n \tag{3.2}$$

$$c_n = \sum_{k=0}^{n} a_k b_{n-k} \tag{3.3}$$

(3.4)

is holonomic. If both a_n and b_n are C-finite sequences, then so is c_n.

Proof. By definition there exist rational functions $q_0, \ldots, q_d \in K(x)$, not all zero, such that

$$a_{n+d} + q_{d-1}(n)a_{n+d-1} + \cdots + q_0(n)a_n = 0, \quad n \geq 0.$$

Repeated use of this recurrence shows that for any $k \geq 0$ fixed the shifted sequence $(a_{n+k})_{n\geq 0}$ can be expressed as linear combination of $(a_n)_{n\geq 0}, \ldots, (a_{n+d-1})_{n\geq 0}$ over the field $K(n)$. In other words, they belong to a K-vector space of dimension at most d. In the same way, the shifted sequences $(b_{n+k})_{n\geq 0}$ for $k \geq 0$ fixed belong to the $K(n)$-vector space spanned by $(b_n)_{n\geq 0}, \ldots, (b_{n+e-1})_{n\geq 0}$.

The proof for each of the sequences (3.1)–(3.3) now follows the same pattern as the following argument for the termwise addition. Let

$$(c_n)_{n\geq 0} = (a_n)_{n\geq 0} + (b_n)_{n\geq 0}.$$

Then the sequence $(c_n)_{n\geq 0}$ belongs to the $K(n)$-vector space generated by

$$(a_n)_{n\geq 0}, \ldots, (a_{n+d-1})_{n\geq 0}, (b_n)_{n\geq 0}, \ldots, (b_{n+e-1})_{n\geq 0},$$

since this vector space contains all the shifted sequences $(a_{n+k})_{n\geq 0}$ and $(b_{n+k})_{n\geq 0}$. The dimension of this space is at most $d + e$. Hence, any $d + e + 1$ sequences $(c_{n+k})_{n\geq 0}$ must be linearly dependent. □

Both, the translation between recurrence relation of a holonomic sequence to the differential equation of a holonomic function, and the proof of the closure properties above are constructive and can be turned into an algorithm. These algorithms are implemented in different packages, such as GeneratingFunctions [15] or HolonomicFunctions [12] in Mathematica or gfun [18] in Maple. We illustrate how the recurrence of the sum of two sequences can be computed with a simple example.

Example 3.7. *Define $q_n(x) = T_n(x) + p_n(x)$, where $T_n(x)$ denote the Chebyshev polynomials of the first kind defined by*

$$T_{n+2}(x) - 2xT_{n+1}(x) + T_n(x) = 0, \quad T_0(x) = 1,\ T_1(x) = x,$$

and $p_n(x) = (1+x)^n$, i.e., the sequence defined by

$$p_{n+1}(x) - (1+x)p_n(x) = 0, \quad p_0(x) = 1.$$

Chebyshev polynomials satisfy a difference equation of order 2 and the polynomials $p_n(x)$ satisfy a difference equation of order 1. Hence, any $2+1+1=4$ shifts of $q_n(x)$ have to be linearly dependent. Thus there has to exist an equation of the form

$$c_3 q_{n+3}(x) + c_2 q_{n+2}(x) + c_1 q_{n+1}(x) + c_0 q_n(x) = 0.$$

Plugging in the definition of $q_n(x)$ and using the defining recurrences for shifts of $T_n(x)$ and $p_n(x)$ we obtain the following system of equations for the shifts of q_n:

$$\begin{aligned} q_{n+3}(x) &= (4x^2-1)T_{n+1}(x) - 2xT_n(x) + (1+x)^3 p_n(x) \\ q_{n+2}(x) &= 2xT_{n+1}(x) - T_n(x) + (1+x)^2 p_n(x) \\ q_{n+1}(x) &= T_{n+1}(x) + (1+x)p_n(x) \\ q_n(x) &= T_n(x) + p_n(x). \end{aligned}$$

Thus the coefficients (c_0, c_1, c_2, c_3) have to appear in the nullspace of the matrix

$$\begin{pmatrix} 4x^2-1 & 2x & 1 & 0 \\ -2x & -1 & 0 & 1 \\ (1+x)^3 & (1+x)^2 & 1+x & 1 \end{pmatrix}.$$

This nullspace can be computed and readily we obtain that

$$c_0(x) = 1+x, \quad c_1(x) = -1-2x-2x^2, \quad c_2(x) = 1+3x, \quad \textit{and} \quad c_3(x) = -1.$$

An immediate consequence of the closure properties for holonomic sequences and the definition of holonomicity of generating functions is that given f, g holonomic also $f+g$, fg, $f'(x)$, etc. are holonomic. Implicitly, we have introduced a data structure to represent sequences or functions that allows also to represent the results under certain transformations.

For his master thesis [15], Christian Mallinger implemented the execution of closure properties in the Mathematica package GeneratingFunctions. This package as well as the master thesis are available for download at

`http://www.risc.jku.at/research/combinat/software/`

Using Mallinger's package example 3.7 can be solved using the "REPlus" command. Concerning the syntax it is necessary to specify both given sequences that shall be added using the same parameters and also the output will be given in these terms. After loading the package, we call REPlus with the defining recurrences for $T_n(x)$ and $p_n(x)$ as input and use as common notation for both $q[n]$ (omitting the dependency on x since it is not relevant in our context).

In[1]:= **<< GeneratingFunctions.m**

GeneratingFunctions Package by Christian Mallinger – © RISC Linz – V 0.67 (03/13/03)

In[2]:= **REPlus[$q[n+2] - 2xq[n+1] + q[n] == 0, q[n+1] - (1+x)q[n] == 0, q[n]$]**

Out[2]= $\left(2x^2+2x+1\right) q[n+1] + (-x-1)q[n] + (-3x-1)q[n+2] + q[n+3] == 0$

Closure properties are implemented on both, the sequence and the function level. The corresponding commands are, e.g., REPlus and DEPlus, RECauchy and DECauchy, ... Also switching between difference equation of the coefficient sequence and differential equation of the generating function is possible using the RE2DE or DE2RE commands. We use RE2DE to derive a closed form for the generating function of Legendre polynomials $F(z) = \sum_{n\geq 0} P_n(x)z^n$. As input we give the Legendre three term recurrence (2.18).

In[3]:= **ode = RE2DE[$\{(n+2)p[n+2] - (2n+3)xp[n+1] + (n+1)p[n] == 0, p[0] == 1,$ $p[1] == x\}, p[n], F[z]$]**

Out[3]= $\left\{F[z](-(x-z)) - \left(2xz - z^2 - 1\right) F'[z] == 0, F[0] = 1\right\}$

In[4]:= **DSolve[ode, $F[z], z$]**

Out[4]= $\left\{\left\{F[z] \to \frac{1}{\sqrt{-2xz+z^2+1}}\right\}\right\}$

Using the Mathematica built-in command DSolve we can solve the resulting ordinary differential equation (ODE) and obtain this way the closed form for $F(z)$. Note that for this type of transformations it is advised to specify also the initial values. Not adding this information will lead to an unnecessary increase of the order of the differential equation, because then the lower coefficients are removed by differentiating the whole equation. In the example of the Legendre generating function the computation without initial values of the recurrences works, but returns an ODE of order 3.

Some expressions can be dealt with by a series of closure properties. For instance, if we consider the sum

$$s_n(x) = \sum_{k=0}^{n} \frac{2k+1}{2} P_k(x).$$

Looking back at chapter 2, we find that this are the Legendre kernel polynomials for $y = 1$ and we know by Christoffel-Darboux that a closed form for this sum exists in general. If we define the coefficients $c_k = \frac{2k+1}{2}$ then this sequence is defined by $(2k+1)c_{k+1} - (2k+3)c_k = 0$ and $c_0 = \frac{1}{2}$. A recurrence for the sum can be determined by first computing a recurrence for the Hadamard product $d_k = c_k P_k(x)$ and then computing the partial sum as the Cauchy product of d_k and the constant sequence 1.

In[5]:= **prod = REHadamard[$\{(n+2)s[n+2] - (2n+3)xs[n+1] + (n+1)s[n] == 0,$ $s[0] == 1, s[1] == x\}, \{(2n+1)s[n+1] - (2n+3)s[n] == 0, s[0] == \frac{1}{2}\}, s[n]$]**

Out[5]= $\{(1+n)(5+2n)s[n]-(1+2n)(5+2n)xs[1+n]+(2+n)(1+2n)s[2+n]==0, s[0]==\frac{1}{2}, s[1]==\frac{3x}{2}\}$

In[6]:= **sum = RECauchy[prod, {s[n + 1] − s[n] == 0, s[0] == 1}, s[n]]**

Out[6]= $\{-(2n+7)s[n+1](2nx+n+3x+2)+(2n+3)s[n+2](2nx+n+7x+3)+(n+2)(2n+7)s[n] - (n+3)(2n+3)s[n+3]=0, s[0]=\frac{1}{2}, s[1]=\frac{3x}{2}+\frac{1}{2}, s[2]=\frac{15x^2}{4}+\frac{3x}{2}-\frac{3}{4}\}$

Using closure properties to compute recurrences for compound sequences does not yield the minimal recurrences. In fact in the example above we overshoot already. For many questions it is not essential to have the smallest possible recurrence, e.g., if we are interested in proving a certain identity. However, if we want to use, e.g., the recurrence for fast computation we might prefer the smallest possible recurrence.

One way of finding a smaller recurrence is to *guess* it. Guessing is another feature implemented in GeneratingFunctions. Given a data set, it tries to solve for polynomial coefficients of a difference equation satisfied by the given input. For this purpose an assumed order of the recurrence d and a maximal degree K for the polynomial coefficients is assumed. The data is plugged in the generic ansatz

$$c_0(n)a_n + \dots c_d(n)a_{n+d} = 0, \qquad \text{with} \quad c_i(n) = \sum_{k=0}^{K} \gamma_{i,k} n^k,$$

and we try to solve the resulting system for the coefficients $(\gamma_{i,k})_{i=0,k=0}^{d,K}$. First of all, this system need not have a solution, either in general or because the order and degree bound were chosen to low. Secondly, the result is an equation that is a priori satisfied by the given input only. If we want to use it to describe the given sequence, we need to still prove it. On the other hand, proving is often easier than finding. We continue the example from above and will first guess a shorter recurrence and then verify this guess using the bigger recurrence (that is certified).

In[7]:= **data = Table[Sum[LegendreP[k, x](2k + 1)/2, {k, 0, n}], {n, 0, 30}]//Factor;**
In[8]:= **guess = GuessRE[data, s[n]]**

Out[8]= $\{\{(2+n)(5+2n)s[n]+(-1-15x-16nx-4n^2x)s[1+n]+(2+n)(3+2*n)s[2+n]==0, s[0]==\frac{1}{2}, s[1]==\frac{1+3x}{2}\}, \text{"ogf"}\}$

In order to prove that this is actually the recurrence satisfied by $s_n(x)$, we compute the recurrence of the difference of the guessed sequence and the certified sequence obtained via closure properties. Since any sequence $(a_n)_{n\geq 0}$ and its negated version $(a_n)_{n\geq 0}$ satisfy the same recurrence only with different initial values, we only need to flip the initial values of the guess for computing the difference of the two sequences.

In[9]:= **REPlus[sum, {guess[[1, 1]], s[0] == −1/2, s[1] == −(1 + 3x)/2}, s[n]]**

Out[9]= $\{-((2+n)(7+2n)s[n])+(7+2n)(2+n+3x+2nx)s[1+n]-(3+2n)(3+n+7x+2nx)s[2+n]+ (3+n)(3+2n)s[3+n]==0, s[0]==0, s[1]==0, s[2]==0\}$

Since the initial values are all zero, the difference of the two sequences must be zero and hence they are equal. Thus we have proved the guessed recurrence. For sake of shorter presentation we have already simplified the initial values in the Mathematica output above. The actual output of GeneratingFunctions gives the unfactored initial values. Mallinger's package only deals with univariate sequences and if not specified otherwise using the corresponding option, the default bounds are 2 for the order and 3 for the degree. Often it is necessary to raise these bounds. Manuel Kauers implemented the Mathematica package Guess (also available for download) that can handle also multivariate input.

3.2 Gosper's algorithm

Next we turn to algorithms that deal with summation other than exploiting closure properties. The first method we discuss is Gosper's algorithm [8] that is capable of *deciding* whether a given hypergeometric term can be summed indefinitely to a closed form in terms of a hypergeometric expression. Before we introduce Gosper's algorithm, we briefly go back one level and show how to sum polynomials in closed form, i.e., find $s \in K[x]$ such that given $p \in K[x]$,

$$s(n) = \sum_{k=0}^{n} p(k).$$

In other words, the problems we are addressing in this section are finding antidifferences of given expressions. This problem is very similar to indefinite integration, i.e., if we are given a function $f(x)$ and we know (or can find) $F(x)$ such that $F'(x) = f(x)$, then clearly

$$\int_a^b f(x)\,\mathrm{dx} = F(b) - F(a).$$

In the same way, if we are given a summand $f(k)$ and are able to determine $F(k)$ such that $F(k+1) - F(k) = f(k)$ then by telescoping

$$\sum_{k=a}^{b} f(k) = F(b+1) - F(a).$$

In the following we will frequently denote the forward shift with respect to a variable k by S_k, i.e., $S_k F(k) = F(k+1)$, and the forward difference by $\triangle_k F(k) = F(k+1) - F(k)$. If the variable is clear from the context, we simply write $\triangle = \triangle_k$ or $S = S_k$.

The first observation concerning solving the problem of finding the antidifference for polynomials is the analogy of forward difference operating on falling factorials to differentiating monomials. For $d \geq 0$ we have

$$\begin{aligned}
\triangle n^{\underline{d}} &= (n+1)^{\underline{d}} - n^{\underline{d}} \\
&= (n+1)n \cdot \dots \cdot (n+1-d)(n+1-d+1) - n(n-1) \cdot \dots \cdot (n-d+1) \\
&= n(n-1) \cdot \dots \cdot (n-d+2)\left((n+1) - (n-d+1)\right) \\
&= d n^{\underline{d-1}}.
\end{aligned}$$

A simple consequence is that

$$\sum_{k=0}^{n} k^{\underline{d}} = \sum_{k=0}^{n} \frac{1}{d+1} \triangle k^{\underline{d+1}} = \frac{1}{d+1}(n+1)^{\underline{d+1}}.$$

Since any polynomial can be expressed in the basis of falling factorials a sum over a polynomial summand can be evaluated by a basis transfer and repeated application of the summation identity above. As a by-product, this also shows that if $s(n) = \sum_{k=0}^{n} p(k)$, where p is a polynomial of degree d, then s is a polynomial of degree $d+1$. This gives another way of evaluating sums over polynomials, namely by interpolation.

The problem we address next is:

When is the sum over a hypergeometric term again hypergeometric?

As opposed to polynomials for hypergeometric input the sum need not always be hypergeometric again. The harmonic numbers defined as

$$H_n = \sum_{k=1}^{n} \frac{1}{k}$$

are a simple counter example. The summand is rational (and as such a special case of a hypergeometric term), the harmonic numbers however are neither rational nor hypergeometric. Let in the following f_k be a given hypergeometric term, i.e., there exists a rational function u such that $f_{k+1} = u(k) f_k$ (wherever $u(k)$ is defined). The output of our consideration will be either a rational function w such that $s_k = w(k) f_k$ satisfies the *telescoping equation*

$$f_k = s_{k+1} - s_k, \tag{3.5}$$

and as such solves our summation problem, or the answer "NO hypergeometric solution to the telescoping problem (3.5) exists". Again, note that this is a *decision procedure*. If a hypergeometric solution exists, it can be determined by Gosper's algorithm. First of all, it is easy to verify that if a hypergeometric solution to (3.5) exists, then it has to be a rational multiple of the given summand, since

$$\frac{s_k}{f_k} = \frac{s_k}{s_{k+1} - s_k} = \frac{1}{\frac{s_{k+1}}{s_k} - 1}$$

which is the rational function w we are looking for. Dividing the telescoping equation through f_k we arrive at the new problem of finding $w \in K(x)$ such that

$$1 = w(k+1)u(k) - w(k). \tag{3.6}$$

So we already reduced the problem of finding a hypergeometric solution to (3.5) to finding a rational solution to (3.6). And we are going to reduce further. The key is to determine a *Gosper form* for the given u:

$$u(x) = \frac{p(x+1)}{p(x)} \frac{q(x)}{r(x+1)}, \tag{3.7}$$

where $p, q, r \in K[x]$ and q, r are such that

$$\gcd\left(q(x), r(x+j)\right) \in K, \qquad j \in \mathbb{N}^*. \tag{3.8}$$

This Gosper form is not uniquely defined, but any representation satisfying the above constraints will do the job as we will see below. First we show how to compute such a factorization.

We start by representing $u(x) = \frac{s(x)}{t(x)}$ with s, t being two relatively prime polynomials. If s, t satisfy the shift condition (3.8) then we are done and

$$q(x) = s(x), \quad r(x) = t(x-1), \quad \text{and} \quad p(x) = 1.$$

In order to check whether (3.8) is satisfied (or to find the values for which it is violated), we use resultants. Given two polynomial p, q with full factorization

$$p(x) = (x - r_1) \cdot \dots \cdot (x - r_m) \quad \text{and} \quad q(x) = (x - s_1) \cdot \dots \cdot (x - s_n),$$

we can define the resultant $\text{res}_x(p, q)$ as

$$\text{res}_x(p, q) = \prod_{i=1}^{m} \prod_{j=1}^{n} (r_i - s_j),$$

i.e., as the evaluation of one polynomial at the roots of the other one. Resultants are implemented in all major computer algebra systems. Compute the resultant

$$R(j) = \text{res}_x \left(s(x), t(x+j) \right).$$

$R(j)$ is a polynomial such that $R(\tilde{j}) = 0$ if and only if $\gcd\left(s(x), t(x+\tilde{j})\right)$ is a non constant polynomial. Thus the positive integer roots of R are precisely the values violating (3.8). These factors $x - \tilde{j}$ are factored out of s and t and put into p (possibly including compensating factors). The procedure can be iterated until we arrive at a Gosper form of u.

Currently we are looking for a rational w satisfying (3.6). We refine the ansatz for w and search for $y \in K(x)$ such that

$$w(k) = \frac{r(k)}{p(k)} y(k)$$

is a solution. If we plug into (3.6) we arrive at the *Gosper equation*

$$p(k) = q(k) y(k+1) - r(k) y(k). \tag{3.9}$$

"And now a miracle happens" [14]. If a rational function y satisfies (3.9), then y has to be polynomial.

Assume y was a rational function and write it as $y(k) = \frac{s(k)}{t(k)}$ with s, t being relatively prime and t being a nontrivial polynomial. If we plug into Gosper's equation then it reads as

$$p(k) t(k) t(k+1) = q(k) s(k+1) t(k) - r(k) s(k) t(k+1).$$

Let $N \in \mathbb{N}$ be the largest integer such that $\gcd\left(t(k), t(k+N)\right) \notin K$. Then there exists a non constant, irreducible polynomial x dividing the greatest common divisor. But then also

$$x(k-N) \mid t(k) \quad \Rightarrow \quad x(k-N) \mid r(k) s(k) t(k+1).$$

Since s, t are relatively prime and $x(k-N) \mid t(k+1)$ would violate the maximality of N we must have $x(k-N) \mid r(k)$ or, shifted forward, $x(k+1) \mid r(k+N+1)$. Analogously it can be argued that $x(k+1) \mid q(k) s(k+1) t(k)$ which implies that $x(k+1) \mid q(k)$. But then we have found a non-trivial factor x for which

$$x(k+1) \mid \gcd\left(q(k), r(k+N+1)\right),$$

which contradicts condition (3.8) of the Gosper form.

This way finally our task turned into determining a *polynomial* solution y to

$$p(x) = q(x)y(x+1) - r(x)y(x). \tag{3.10}$$

with given r, q, p. Based on the polynomial degrees of r, q and p a degree bound d for y can be computed:

If $\deg q \neq \deg r$ or $\mathrm{lc}(q) \neq \mathrm{lc}(r)$ then

$$\mathcal{D} = \{\deg(p) - \max\{\deg(q), \deg(r)\}\}. \tag{3.11}$$

Else, with $q_{k-1} = \langle x^{k-1}\rangle q(x)$ and $r_{k-1} = \langle x^{k-1}\rangle r(x)$:

$$\mathcal{D} = \{\deg(p) - \deg(a) + 1, (r_{k-1} - q_{k-1})/\operatorname{lc}(a)\}\,. \tag{3.12}$$

If we intersect $\mathcal{D}$ with the natural numbers, then if $\mathcal{D} = \emptyset$ then no polynomial solution exists (and thus no hypergeometric solution exists!). Otherwise put $d = \max \mathcal{D}$. Plugging in an ansatz for y up to this degree in the Gosper equation (3.10) we can solve for the undetermined coefficients. If no polynomial solution exists, then no hypergeometric solution exists to the original problem. Otherwise, return

$$w(n) = \frac{r(n)}{p(n)}y(n).$$

We summarize the main steps of **Gosper's algorithm:**
Input: $(f_k)_{k\geq 0}$ hypergeometric

Output: $(s_k)_{k\geq 0}$ hypergeometric such that $s_k = f_{k+1} - f_k$ OR "impossible" if no such s_k exists.

1. Let $u \in K(x)$ be: $u(k) = \frac{f_{k+1}}{f_k}$
2. Compute the *Gosper form* of u:

$$u(x) = \frac{p(x+1)}{p(x)}\frac{q(x)}{r(x+1)},$$

   with $p, q, r \in K[x]$ and q, r such that $\gcd(q(x), r(x+j)) \in K$ for all $j \geq 1$
3. Find a polynomial solution $y \in K[x]$ of *Gosper's equation*

$$p(x) = q(x)y(x+1) - r(x)y(x)$$

   (using the degree bounds (3.11) and (3.12)). If no such y exists, return "impossible"
4. Let $w(x) = \frac{r(x)}{p(x)}y(x)$
5. Return $s_k = w(k)f_k$

We illustrate the execution of Gosper's algorithm with two simple examples, the well known Gauß-sum and the harmonic numbers (the latter to prove that harmonic numbers are not hypergeometric).

Example 3.8. *Let $F_n = \sum_{k=1}^{n} k$ for $n \geq 1$, i.e., we have $f_k = k$. We follow the steps of Gosper's algorithm:*

1. *compute the shift quotient $u(k) = \frac{f_{k+1}}{f_k} = \frac{k+1}{k}$ (defined for $k \geq 1$)*
2. *determine the Gosper form of u: in this case it is easy to be seen that the choice*

$$q(x) = r(x) = 1, \quad p(x) = x$$

   does the job.

3. *Find a polynomial solution y to Gosper's equation:*

$$x = y(x+1) - y(x).$$

   For the degree bound, we run into the first part of case (3.12), i.e., $d = \deg p(x) - \deg q(x) + 1 = 1 - 0 + 1 = 2$. Hence we use the ansatz $y(x) = y_2 x^2 + y_1 x + y_0$. Plugging into Gosper's equation gives

$$x = 2xy_2 + y_1 + y_2,$$

   and equating the coefficients of x of like powers on both sides yields the system

$$\begin{array}{rcl} 1 & = & 2y_2 \\ 0 & = & y_1 + y_2 \end{array} \qquad \Rightarrow \qquad y_2 = \tfrac{1}{2}, \quad y_1 = -\tfrac{1}{2}.$$

   Thus $y(x) = \frac{1}{2}x(x-1) + y_0$

4. *$w(x) = \frac{1}{2}(x-1)$*
5. *Return $s_k = \frac{1}{2}(k-1)k$*

This yields the well-known result $F_n = \sum_{k=1}^{n}(s_{k+1} - s_k) = s_{n+1} - s_1 = \frac{n(n+1)}{2}$.

Example 3.9. *Let $H_n = \sum_{k=1}^{n} \frac{1}{k}$ for $n \geq 1$, i.e., $f_k = \frac{1}{k}$:*

1. *compute the shift quotient $u(k) = \frac{f_{k+1}}{f_k} = \frac{k}{k+1}$*
2. *determine the Gosper form of u: in this case it is easy to see that the choice*

$$q(x) = r(x) = x, \quad p(x) = 1$$

   does the job.

3. *Find a polynomial solution y to Gosper's equation:*

$$1 = xy(x+1) - xy(x).$$

   For the degree bound again by (3.12) it follows that $d = 0$, i.e., $y(x) = y_0$. Plugging into Gosper's equation gives

$$1 = xy_0 - xy_0 = 0,$$

   which does not have a solution. I.e. we return "NO hypergeometric solution exists"

There exist several implementations of Gosper's algorithm, e.g., in Mathematica the package zb.m implemented by Peter Paule and Markus Schorn [16] that is also available for download on the RISC algorithmic combinatorics software page.

In[10]:= **<< zb.m**

Fast Zeilberger Package by Peter Paule and Markus Schorn (enhanced by Axel Riese) – © RISC Linz – V 3.54 (02/23/05)

In[11]:= **Gosper[*kk*!, {*k*, 0, *n*}]**

If 'n' is a natural number, then:

Out[11]= $$\left\{\sum_{k=0}^{n} kk! == (n+1)n! - 1\right\}$$

In[12]:= $$\mathbf{Gosper}\left[\frac{4k-1}{(2k-1)^2}16^{-k}\binom{2k}{k}^2, \{k, 0, n\}\right]$$

If 'n' is a natural number, then:

Out[12]= $$\left\{\sum_{k=0}^{n} \frac{4k-1}{(2k-1)^2}16^{-k}\binom{2k}{k}^2 == -16^{-n}\binom{2n}{n}^2\right\}$$

Entering the command "Prove[ ]" returns the proof of the sum evaluation provided by Gosper's algorithm, i.e., it gives the telescoper that yields the closed form, in human readable form.

3.3 Zeilberger's algorithm

In the previous section we were dealing with the problem of indefinite summation. Often it is too much to ask for an antidifference of a given (hypergeometric) term, still there exists a closed form evaluation of the *definite* sum. A simple example is

$$\sum_k \binom{n}{k} = \sum_{k=0}^{n} \binom{n}{k} = (1+1)^n = 2^n.$$

In the first sum (and also in the following) we consider the summation variable to range over all integers. Since $\binom{n}{k} = 0$ if $k < 0$ or $k > n$ this summation is in fact finite. By the binomial theorem we know that there exists a simple closed form evaluation. But for the definite summation

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

no such (hypergeometric) closed form exists. The input we consider for now is a summand $f(n,k)$ that is hypergeometric in both n and k and we are interested in the definite sum

$$s(n) = \sum_k f(n,k)$$

with k ranging over all integers. Following the idea of telescoping, if we would find $g(n,k)$ such that

$$g(n,k+1) - g(n,k) = f(n,k),$$

then we would have a hold on $s(n)$. This might still be too much to ask for. Hence we extend the idea following Zeilberger's approach [26, 25] and search for polynomial coefficients $c_0(n), \dots, c_d(n)$ (not all zero) such that

$$c_0(n)f(n,k) + c_1(n)f(n+1,k) + \dots c_d(n)f(n+d,k) = g(n,k+1) - g(n,k). \qquad (3.13)$$

This idea is also referred to as *creative telescoping*. Note that the coefficients above do not depend on the summation variable k. Hence by summing over (3.13) a (possibly inhomogeneous) recurrence for $s(n)$ is obtained.

Verbaeten [21, 20] showed that for *proper* hypergeometric summands $f(n,k)$ a k-free recurrence

$$\sum_{i=0}^{I}\sum_{j=0}^{J} a_{ij}(n)f(n+i,k+j) = 0 \qquad (3.14)$$

exists and he also gave explicit bounds on the orders I and J, see also [23]. For the proof of the order bounds it is necessary to require that the summand is proper hypergeometric, i.e., that the shift ratios with respect to both n and k split into integer linear factors. We denote the shift quotients by

$$r(n,k) = \frac{f(n+1,k)}{f(n,k)} \quad \text{and} \quad s(n,k) = \frac{f(n,k+1)}{f(n,k)}. \qquad (3.15)$$

With this notation, the shift quotients $f(n+i,k+j)/f(n,k)$ can be expressed as products of shifts of r and s, e.g.,

$$\frac{f(n+2,k)}{f(n,k)} = \frac{f(n+2,k)}{f(n+1,k)} \cdot \frac{f(n+1,k)}{f(n,k)} = r(n+1,k)r(n,k).$$

Hence we can divide the equation (3.14) through $f(n,k)$ and obtain a linear combination of products of shifts of r and s. The next step is to bring this equation to a common denominator and equate the coefficients of powers of k in the numerator to zero. If there are more unknowns $a_{ij}(n)$ than equations then a nontrivial solution will exist. The number of unknowns is obviously $(I+1)(J+1)$. Now if the summand is proper hypergeometric then the degree of the numerator with respect to k grows linearly in I and J. At some point the number of unknowns will exceed the number of equations and then at latest we have a nontrivial nullspace of the system. For instance consider the summand

$$f(n,k) = \frac{(2n+k+1)!}{(n+2k+1)!} \quad \Longrightarrow \quad \begin{cases} r(n,k) & = & \frac{(2n+k+2)(2n+k+3)}{n+2k+2} \\ \text{and} & & \\ s(n,k) & = & \frac{2n+k+2}{(n+2k+2)(n+2k+3)}. \end{cases}$$

Then for $I = J = 1$ the degree of k in the numerator is 4 and also the number of unknowns is $(I+1)(J+1) = 4$. For $I = J = 2$, the degree is 8 which gives rise to 9 equations and the number of unknowns is 9. For, e.g., $I = 3$ and $J = 2$ the degree is 10 and the number of unknowns is 12 and we can stop iterating and solve the linear system for the coefficients $a_{ij}(n)$.

On the other hand, if the shift quotient contains an arbitrary polynomial factor in the denominator then the number of equations (i.e., the degree of k in the numerator) grows too fast. A classical example where this is the case is the summand

$$f(n,k) = \frac{1}{n^2+k^2}.$$

Below you find a table with the number of equations and unknowns for different choices of I and J:

I	1	1	1	2	2	2	3	3	3
J	1	2	3	1	2	3	1	2	3
No(equations)	6	10	14	10	16	22	14	22	30
No(unknowns)	4	6	8	6	9	12	8	12	16

Given the existence of a k-free recurrence (3.14) it can be shown that a creative telescoping identity of the form (3.13) exists.

Theorem 3.10. *Let $f(n,k)$ be a proper hypergeometric term. Then f satisfies a nontrivial recurrence of the form* (3.13) *in which $g(n,k)/f(n,k)$ is a rational function of n and k.*

Proof. See e.g. [14]. □

The theorem above guarantees the existence of a relation

$$c_0(n)f(n,k) + c_1(n)f(n+1,k) + \dots c_d(n)f(n+d,k) = g(n,k+1) - g(n,k)$$

with g being a rational multiple of $f(n,k)$. The question is now how to determine such a recurrence in a reasonably fast way. This leads us back to Gosper's algorithm. First fix the assumed order d of the recurrence, e.g., $d = 1$. Then we seek to compute a telescoper g of

$$h(n,k) = c_0(n)f(n,k) + c_1(n)f(n+1,k) = (c_0(n) + c_1(n)r(n,k))\, f(n,k),$$

in the notation introduced above. Following Gosper we first determine the shift quotient

$$u(n,k) = \frac{h(n,k+1)}{h(n,k)} = \frac{c_0(n) + c_1(n)r(n,k+1)}{c_0(n) + c_1(n)r(n,k)} s(n,k),$$

and then the Gosper form of $u(n,k)$. From the Gosper form we can set up the Gosper equation which now needs to be solved for a polynomial solution y and the coefficients c_0, c_1. The presence of these coefficients gives an additional freedom for a solution to exist (even though classical Gosper would not work on the given summand). If no polynomial solution exists, then the order d is increased and we start all over. This process has to terminate by the considerations of Verbaeten.

A Maple implementation of Zeilberger's algorithm can be found, e.g., on Zeilberger's homepage. A Mathematica implementation is part of the zb-package introduced in the previous section. For the binomial sum discussed in the beginning it is used as follows:

```
In[13]:= Zb[Binomial[n, k], {k, 0, n}, n]
    If 'n' is a natural number, then:

Out[13]= {2SUM[n] - SUM[1 + n] == 0}
```

If the resulting recurrence is as simple as, e.g., in this case then a closed form evaluation of the sum is possible. Also if the recurrence is of higher order, but with constant coefficients (C-finite) a closed form solution can be easily determined. In any case a (possibly inhomogeneous) recurrence is returned.

The summand in the sum representation (2.20) of Jacobi polynomials is also proper hypergeometric in n and k and thus Zeilberger's algorithm is applicable to derive the Jacobi three term recurrence:

In[14]:= $\mathbf{Zb}\left[\frac{(\alpha+1)_n}{n!}\,\frac{(-n)_k(n+\alpha+\beta+1)_k}{(\alpha+1)_k\;k!}\left(\frac{1-x}{2}\right)^k,\{k,0,n\},n\right]$

Out[14]= $\{-2(n+\alpha+1)(n+\beta+1)(2n+\alpha+\beta+4)\mathrm{SUM}[n]$
$+(2n+\alpha+\beta+3)\left((2n+\alpha+\beta+2)(2n+\alpha+\beta+4)x+(\alpha^2-\beta^2)\right)\mathrm{SUM}[1+n]$
$-2(n+2)(n+\alpha+\beta+2)(2n+\alpha+\beta+2)\mathrm{SUM}[2+n]==0\}$

The same recurrence is also found starting from the sum representation given in theorem 2.13, which can be derived from the Rodrigues formula by application of Leibniz rule. For the special case $\alpha=\beta=0$, we have the three term recurrence for Legendre polynomials in an easy and fast way. Later we will see an extension of creative telescoping that also allows to *find* (and thus also *prove*) differential relations or mixed difference-differential relations as given in theorem 2.9.

3.4 A short introduction to Gröbner bases

Gröbner bases were originally developed by Bruno Buchberger [3] to solve the problem of ideal membership. Since they were introduced they have successfully been applied to numerous problems ,some of which will be mentioned in this section and used later on. Here we will only give a very brief introduction, for more complete informations we refer to [24] and references therein.

First we recall some common notations and definitions. In the following we will frequently abbreviate $K[x_1,\ldots,x_n]=K[X]$. By $[X]$ we denote the monoid (under multiplication) of power products $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$. A subset $I\subset K[X]$ is called an ideal over the polynomial ring $K[X]$, iff for any $p,q\in I$ also the sum $p+q\in I$ and if for any $p\in I$ and $r\in K[X]$ the product $p\cdot r\in I$. We write $I\trianglelefteq K[x]$.

A set of polynomials $\{b_1,\ldots,b_k\}$ forms a basis of I, if

$$I=\{p_1b_1+\cdots+p_kb_k\ \mid\ p_1,\ldots,p_k\in K[X]\}.$$

We write $I=\langle b_1,\ldots,b_k\rangle$. In the univariate case it is obvious which coefficient is the leading coefficient (and which term is the leading term). In the multivariate case we need to specify a term order in order to clarify the notion of a leading term.

Definition 3.11. *A* term order *is a total order* $\prec$ *satisfying*

- $\forall\,\tau\in[X]:\ 1\prec\tau$
- $\forall\,\tau_1,\tau_2,\sigma\in[X]:\ \tau_1\prec\tau_2\Rightarrow\tau_1\sigma\prec\tau_2\sigma$

In the multivariate setting there are several possible term orderings. We list only a few for $K[x,y]$, the generalization to $K[X]$ is straight forward.

1. *lexicographic ordering:* First fix an order on the variables, e.g. $x>y$. Then $x^{i_1}y^{j_1}\prec_{lex}x^{i_2}y^{j_2}$ iff either $i_1<i_2$ or, if $i_1=i_2$ and $j_1<j_2$.

2. *graduated lexicographic ordering:* Again let $x > y$. First we go by the total degree of the term and if this is equal, then we compare using lexicographic ordering, i.e.,

$$x^{i_1}y^{j_1} \prec_{glex} x^{i_2}y^{j_2} \Leftrightarrow i_1 + j_1 < i_2 + j_2 \ \vee \ \left(i_1 + j_1 = i_2 + j_2 \ \wedge \ x^{i_1}y^{j_1} \prec_{lex} x^{i_2}y^{j_2}\right).$$

3. *graduated reverse lexicographic ordering:*

$$x^{i_1}y^{j_1} \prec_{grlex} x^{i_2}y^{j_2} \Leftrightarrow i_1 + j_1 < i_2 + j_2 \ \vee \ \left(i_1 + j_1 = i_2 + j_2 \ \wedge \ x^{i_2}y^{j_2} \prec_{lex} x^{i_1}y^{j_1}\right).$$

For instance, with $x < y$, we have for $\tau_1 = x^2y^3$ and $\tau_2 = x^3y$ that

$$\tau_1 \prec_{lex} \tau_2, \quad \tau_2 \prec_{glex} \tau_1 \quad \text{and} \quad \tau_2 \prec_{grlex} \tau_1.$$

From now on we choose a term order $\prec$ and fix it. The leading term of a polynomial $p \in K[X]$ is the term of p which is maximal with respect to $\prec$ and is denoted by $\mathrm{lt}(p)$. The leading coefficient is the coefficient of the leading term and is denoted as usual by $\mathrm{lc}(p)$.

Next recall polynomial division in the univariate case. Given two polynomials $a, b \in K[x]$ there exist *uniquely* determined polynomials q, r such that

$$a(x) = b(x)q(x) + r(x) \quad \text{with} \quad \deg(r) < \deg(b).$$

For multivariate polynomials in general quotient and remainder are not uniquely defined, unless we specify a term order and replace the degree reduction condition by a divisibilty condition on leading terms. Then we have that given two polynomials $a, b \in K[X] = K[x_1, \ldots, x_n]$ there exist *uniquely* determined polynomials $q, r \in K[X]$ such that

$$a(x_1, \ldots, x_n) = b(x_1, \ldots, x_n)q(x_1, \ldots, x_n) + r(x_1, \ldots, x_n)$$

such that $\mathrm{lt}(b)$ divides *no* term of r. Given $b \in K[X]$ we define the reduction relation $\longrightarrow_{\{b\}}$ by

$$a \longrightarrow_{\{b\}} r \quad :\Leftrightarrow \quad \exists\, q : a = bq + r \quad \text{and} \quad \mathrm{lt}(r) \preceq \mathrm{lt}(a).$$

The reduction with respect to a set of polynomials $\{b_1, \ldots, b_k\}$ is defined as

$$a \longrightarrow_{\{b_1,\ldots,b_k\}} r \quad :\Leftrightarrow \quad \exists\, q_1, \ldots, q_k : a = b_1q_1 + \cdots + b_kq_k + r \quad \text{and} \quad \mathrm{lt}(r) \preceq \mathrm{lt}(a).$$

Note that if $a \longrightarrow_{\{b_1,\ldots,b_k\}} r$ then $a - r \in \langle b_1, \ldots, b_k\rangle$. The reduction with respect to the set $B = \{b_1, \ldots, b_k\}$ is unique (i.e., the remainder r is uniquely determined) if B is a Gröbner basis.

Definition 3.12. *A set $G = \{g_1, \ldots, g_k\} \subset K[X]$ forms a* Gröbner basis *iff*

$$\forall\, a \in \langle g_1, \ldots, g_k\rangle : \quad a \longrightarrow_G 0.$$

If the Gröbner basis is known then we simply write $a \longrightarrow r$ instead of $a \longrightarrow_G r$. There are many equivalent characterizations of Gröbner bases. One of them states that if $a \longrightarrow r_1$ and $a \longrightarrow r_2$ then there exists a polynomial r such that both $r_1 \longrightarrow r$ and $r_2 \longrightarrow r$. This property is referred to as *confluence*.

Buchberger derived an algorithm that computes given a set $\{a_1, \ldots, a_m\} \subset K[X]$ computes a Gröbner basis $G = \{g_1, \ldots, g_k\}$ of $\langle a_1, \ldots, a_m\rangle$. This algorithm gives a constructive proof of the *existence* of a Gröbner basis for every ideal I in $K[X]$. Furthermore, if for any $g \in G$ we have that g is irreducible with respect to $G\backslash\{g\}$ (autoreduced) and if we normalize the elements of G to be monic, then the Gröbner basis is *uniquely determined*.

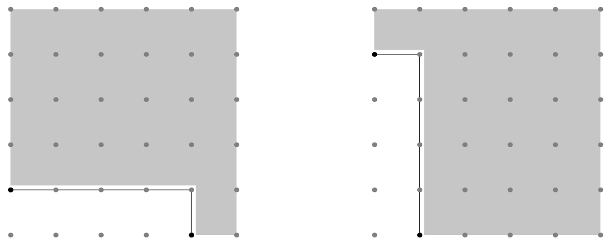
Example 3.13. *Let* $I = \langle p_1(x,y) = x^2 + y^2 - 4, p_2(x,y) = (x-1)(y-1) - 1 \rangle \trianglelefteq K[x,y]$ *and let* $\prec = \prec_{lex}$ *with* $x < y$*. Then the Gröbner basis* G_1 *of* I *is given by*

$$G_1 = \left\{x^4 - 2x^3 - 2x^2 + 8x - 4, x^3 - x^2 - 3x + y + 4\right\}.$$

On the other hand, if we choose $\prec = \prec_{lex}$ *with* $y < x$ *then the Gröbner basis* G_2 *of* I *is given by*

$$\left\{y^4 - 2y^3 - 2y^2 + 8y - 4, x + y^3 - y^2 - 3y + 4\right\}.$$

We can represent a Gröbner basis graphically in the integer lattice $\mathbb{Z}^2$ *with coordinates* (i,j) *for the term* $x^i y^j$ *plotting the leading terms of the basis polynomials. Below you see the corresponding plot for the Gröbner bases* G_1 *and* G_2*.*



Note that all three sets of polynomials $\{p_1, p_2\}, G_1, G_2$ have the same common roots. Observe also that G_1 contains a polynomial only depending on x (i.e., on the smallest variable) and G_2 contains a univariate polynomial in y. This property is referred to as elimination property, see below. Next we give a short and by no means complete list of problems that can be solved using Gröbner basis:

Elimination property In general we have that if G is a Gröbner basis with respect to the lexicographic order $x_1 < \cdots < x_n$ then (with $k < n$)

$$I \cap K[x_1, \ldots, x_k] = \langle G \cap K[x_1, \ldots, x_k] \rangle.$$

This also gives a way to determine the common roots of a given system of polynomials. After computing the Gröbner basis, we can compute the roots of the univariate polynomials in x_1. Next we plug in the solutions and can determine the solutions for x_2 in $I \cap K[x_1, x_2]$ and iterate further.

Ideal membership Given an ideal $I \trianglelefteq K[X]$ and a polynomial $p \in K[X]$ we want to determine wheter $p \in I$. We compute a Gröbner basis G of I and reduce p with respect to G, i.e., $p \longrightarrow_G r$. If $r = 0$, then $p \in I$, otherwise not. Note that also if r does not vanish then it is a *normal form* of the given polynomial in the ideal I.

Equality of ideals Given two ideals, we compute their Gröbner bases with respect to some fixed ordering. If the (normalized, autoreduced) Gröbner are the same, then so are the ideals. Note that if a Gröbner basis of an ideal I consists only of the element $\{1\}$, then this is equivalent to $I = K[X]$.

Ideal dimension The vector space dimension of the factor ring $K[x,y]/_{\langle G\rangle}$ can be read off the number of terms below the "staircase". For the ideal dimension we need the notion of the Hilbert polynomial. Given an ideal $I \trianglelefteq K[X]$ let the Hilbert function $H : \mathbb{Z} \to \mathbb{Z}$ denote the number of irreducible terms of total degree at most d. The Hilbert polynomial is the polynomial $h(d)$ that equals $H(d)$ for d large enough. The ideal dimension is then given by the degree of the Hilbert polynomial. The dimension of $\langle G_k\rangle$ $(k = 1, 2)$ of the example above is thus zero.

Radical membership Given an ideal $I \trianglelefteq K[X]$ and a polynomial $p \in K[X]$, determine whether $p \in \mathrm{Rad}(I)$, where $\mathrm{Rad}(I)$ is the radical ideal defined by

$$\mathrm{Rad}(I) = \sqrt{I} = \left\{a \mid a^k \in I\right\}.$$

The key for invoking Gröbner basis is Hilbert's Nullstellensatz. It asserts that if we are working over an algebraically closed field, then $\mathrm{Rad}(I)$ consists of exactly those polynomials in $K[X]$ vanishing on all common roots of I. If $I = \langle a_1, \ldots, a_m\rangle$, then we compute a Gröbner basis G of $\langle a_1, \ldots, a_m, py - 1\rangle$ in $K[X, y]$, i.e., we introduce a new variable y. If $G = \{1\}$, then p is in the radical ideal, otherwise not. This approach is usually referred to as the Rabinowitsch trick.

3.5 Multivariate holonomic functions

The algorithms for executing closure properties or dealing with sums over (proper) hypergeometric expressions that we have considered so far have been univariate or bivariate at most. Next we want to extend these concepts to holonomic functions in the multivariate case. For this discussion we will use operator notation.

The operators that we are dealing with in this lecture are shifts and derivatives. For forward shifts with respect to the variable n we use the notation S_n, i.e., $S_n f(n) = f(n + 1)$. For derivatives with respect to the variable x we use the notation D_x. Note that these operators are non-commutative, i.e.,

$$D_x x = xD_x + 1 \quad \text{and} \quad S_n n = (n + 1)S_n.$$

But certainly the operators D_x and S_n commute, i.e., $D_xS_n = S_nD_x$. Sometimes we use the symbol ∂_z to denote either of these operators if the statement applies to both.

The input that we consider are multivariate functions depending on discrete and/or continuous variables that satisfy linear difference equations or differential equations or mixed difference-differential equations with polynomial coefficients. Given such functions we discuss an algorithm to carry out closure properties, i.e., operations such as addition, multiplication, and to deal with definite summation and definite integration. The output of this procedure will be a description again in terms of linear difference equations or differential equations or mixed difference-differential equations with polynomial coefficients.

Before we turn to the multivariate case we review first univariate holonomic functions (and sequences) from the operator point of view. In the following we will abuse notation in the sense that we do not use notation that accounts for the non-commutativity or the free algebra structure and instead keep it short and simple. The symbols " $\circ$ " and " $\bullet$ " denote composition and application of an operator, respectively.

Let $A = \mathbb{Q}(z)[\partial_z]$ and let f be a given function. Then define

$$I = \{L \in A \mid L \bullet f = 0\}.$$

Obviously for two operators $L_1, L_2 \in I$ also $L_1 + L_2 \in I$ and for $U \in A$ and $L \in I$ also $U \circ L \in I$ since

$$(U \circ L) \bullet f = U \circ (L \bullet f) = 0.$$

In other words, I is an ideal in $(A, \circ, +)$, the *annihilating ideal* of f denoted by $\operatorname{ann}(f)$. Note that A needs to be constructed in a way that for $U, V \in A$,

$$(U \circ V) \bullet f = U \circ (V \bullet f)$$

holds. For instance if $U = D_x$ and $V = x$, then

$$(U \circ V) \bullet f = (D_x x) \bullet f,$$

and

$$U \circ (V \bullet f) = D_x \circ (xf) = f + xD_x f = (1 + xD_x)f.$$

By the map induced by Leibniz rule

$$\Phi : A \to A \quad D_x x \mapsto xD_x + 1$$

we see that both expressions above are indeed equal and in general we can transform any $p \in A$ into its *standard form:*

$$p \mapsto \sum_{i=0}^{d} a_i(x) D_x^i, \quad \text{with} \quad a_i \in \mathbb{Q}(x).$$

Note that this type of construction can be carried out also for shifts, i.e., $A = \mathbb{Q}(n)[S_n]$, or more general derivations ∂_z. Hence we consider $\left(z^\alpha \partial^\beta\right)_{\alpha,\beta}$ as a basis for the normal form representation.

With this notation a univariate function f (or sequence) is holonomic if there exists $L \in A$ such that $L \bullet f = 0$, i.e., in this case the annihilating ideal is generated by a single operator. As in the univariate case we first define holonomicity for continuous variables in the multivariate case.

Definition 3.14. *Let $A = \mathbb{Q}(x_1, \ldots, x_d)[D_{x_1}, \ldots, D_{x_d}]$. Then a function f is* holonomic *if for any $1 \le i \le d$ there exists a nontrivial operator $P_i \in \operatorname{ann}(f)$ of the form*

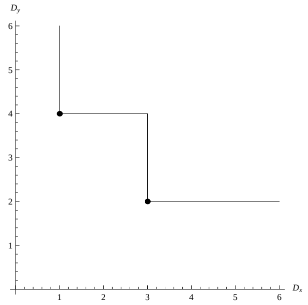
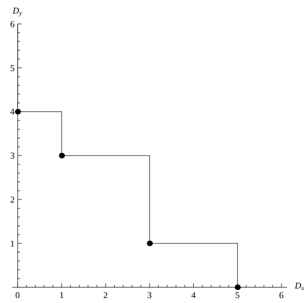
$$P_i = \sum_{j=0}^{m_i} a_{ij}(x_1, \ldots, x_d) D_{x_i}^j$$

with rational function coefficients a_{ij}.

This means that a function is holonomic if $\operatorname{ann}(f)$ contains operators that depend *only* on D_{x_i} for each i. Also in certain non-commutative rings Gröbner bases can be defined and computed, but we will not enter this topic and for sake of simplicity merely use some facts that hold both for the commutative and the non-commutative case.

Note that the Gröbner basis of $\operatorname{ann}(f)$ need not contain any of these univariate operators, but they can be computed using Gröbner bases with the corresponding term order.

If we have a look at the leading power products of a (non-commutative) Gröbner basis of $\operatorname{ann}(f) \trianglelefteq \mathbb{Q}(x,y)[D_x, D_y]$ then in general it will look like in the left picture below. If f is holonomic then the ideal is zero dimensional and the leading power products will look rather like in the right image.

Holonomicity of sequences is again defined via the (multivariate) generating function.

Definition 3.15. *A multivariate sequence* $a(n_1, \ldots, n_d)$ *is* holonomic *if the generating function*

$$F(z_1, \ldots, z_d) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_d=0}^{\infty} a(n_1, \ldots, n_d) z_1^{n_1} \cdots z_d^{n_d}$$

is holonomic.

Holonomic functions that depend on both discrete and continuous variables are then defined as the obvious combination of the above. All orthogonal polynomials that we have discussed so far are examples for such sequences. They involve at least one discrete parameter (the polynomial degree) and one continuous variable.

Example 3.16. *Consider Legendre polynomials* $P_n(x)$ *in* $\mathbb{Q}(n,x)[S_n, D_x]$ *with* $S_n > D_x$ *in lexicographic order. Then the Gröbner basis of* $\operatorname{ann}(P_n(x))$ *is given by*

$$\{(n+1)S_n - (x^2-1)D_x - (n+1)x, (x^2-1)D_x^2 + 2xD_x - n(n+1)\}$$

On the other hand, if we compute the Gröbner basis in $\mathbb{Q}(n,x)[D_x, S_n]$*, i.e.,* $D_x > S_n$*, we obtain*

$$\{(x^2-1)D_x + (-n-1)S_n + (n+1)x, (n+2)S_n^2 - (2n+3)xS_n + (n+1)\}$$

Note that in the first case we have a mixed relation and the Legendre differential equation (2.19) *and in the latter the same mixed relation and the Legendre three term recurrence* (2.18)*.*

Also multivariate holonomic functions are closed under certain operations some of which are listed in the following theorem.

Theorem 3.17. *Let* f, g *be holonomic. Then* $f+g$*,* fg*, definite sums* $\sum f$ *and definite integrals* $\int f$ *are again holonomic.*

As in the univariate case the proof of this theorem is constructive in the sense that given $\operatorname{ann}(f)$ and $\operatorname{ann}(g)$, the annihilating ideal for $f+g,\ldots$ can be computed. Instead of mere linear algebra in the multivariate case this also requires computations of non-commutative Gröbner bases.

These closure properties as well as a higher dimensional version of Zeilberger's algorithm as well as further methods that we do not treat here are implemented in Christoph Koutschan's Mathematica package HolonomicFunctions. For further details on the scope of the program and the underlying algorithms we refer to his thesis [12] or the user's guide [13]. This package is very user friendly as one need not know all the underlying structure and available algorithms, but it is possible to call them with the single command "Annihilator" and internally the right method is chosen. But also the syntax for closure properties, creative telescoping or Gröbner basis reductions is easy to understand and use.

Example 3.18. *The composite sequence $f(n)=\sum_{k\geq 0}\binom{n}{k}P_k(x)$, where at first we consider x as a free parameter and as operations only shifts in n, is holonomic because of closure properties. Applying the "Annihilator" command yields*

In[15]:= **Annihilator[Sum[Binomial[n, k]LegendreP[k, x], {k, 0, n}], {S[n]}]**

Out[15]= $\{(n+2)S_n^2+(-2nx-2n-3x-3)S_n+(2nx+2n+2x+2)\}$

Observe that the input can be specified in classical Mathematica notation and that the shift operator with respect to n is denoted by $S[n]$.

The commands for executing closure properties in HolonomicFunctions are of the form "DFinite[Operation]*". For instance if we want to compute an annihilating ideal for the summand above, we first compute the annihilators for each $P_k(x)$ and $\binom{n}{k}$ and then use "DFiniteTimes":*

In[16]:= **ann1 = Annihilator[LegendreP[k, x], {S[k]}]**

Out[16]= $\{(k+2)S_k^2+(-2kx-3x)S_k+(k+1)\}$

In[17]:= **ann2 = Annihilator[Binomial[n, k], {S[k]}]**

Out[17]= $\{(k+1)S_k+(k-n)\}$

In[18]:= **ann = Factor[DFiniteTimes[ann1, ann2]]**

Out[18]= $\{(k+2)^2S_k^2+(2k+3)x(k-n+1)S_k+(k-n)(k-n+1)\}$

Note that in the last step we wrapped the Mathematica command "Factor" around DFiniteTimes which factors the coefficients of the operators (note that it does not factor the operators!).

Legendre polynomials are also holonomic as functions in x and thus we can consider $f(n)$ also as function $f(n,x)$ and use the operator D_x in addition. Derivation with respect to a variable x is denoted by $Der[x]$ in HolonomicFunctions and readily we obtain

In[19]:= **Factor[Annihilator[Sum[Binomial[n, k]LegendreP[k, x], {k, 0, n}], {S[n], Der[x]}]]**

Out[19]= $\{(-n-1)S_n+2(x-1)(x+1)D_x+(2n+x+1),$
$-2(x-1)(x+1)^2D_x^2+(x+1)(2nx-2n-3x-1)D_x+n(n+x+2)\}$

Example 3.19. *Recall the definition of polynomials orthogonal in the L^2-inner product on triangles defined in section 2.3,*

$$\phi_{i,j}(x,y) = P_i\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i P_j^{(2i+1,0)}(y), \quad i,j \geq 0.$$

Also for these a recurrence relation can be computed using the Annihilator command. In this case it is too big to display here, but there is also a command that allows to have a look at the support of the basis of the annihilating ideal without seeing all the coefficients:

In[20]:= $\mathbf{ann = Factor[Annihilator[LegendreP[}i, \frac{2x}{1-y}\mathbf{]}\left(\frac{1-y}{2}\right)^i \mathbf{JacobiP[}j, 2i{+}1, 0, y\mathbf{], \{}S[i], S[j]\mathbf{\}]];}$

In[21]:= **Support[ann]**

Out[21]= $\{\{S_j^2, S_j, 1\}, \{S_i^2, S_iS_j, S_i, S_j, 1\}\}$

For these basis functions we can also determine the full system of mixed relations including derivatives with respect to x and y:

In[22]:= $\mathbf{ann = Factor[Annihilator[LegendreP[}i, \frac{2x}{1-y}\mathbf{]}\left(\frac{1-y}{2}\right)^i \mathbf{JacobiP[}j, 2i+1, 0, y\mathbf{],}$
$\mathbf{\{}S[i], S[j], Der[x], Der[y]\mathbf{\}]];}$

In[23]:= **Support[ann]**

Out[23]= $\{\{S_j, D_x, D_y, 1\}, \{D_y^2, S_i, D_x, D_y, 1\}, \{D_xD_y, S_i, D_x, D_y, 1\}, \{D_x^2, D_x, 1\}, \{S_iD_y, S_i, D_x, D_y, 1\},$
$\{S_iD_x, S_i, D_x, D_y, 1\}, \{S_i^2, S_i, D_x, D_y, 1\}\}$

The creative telescoping idea of Zeilberger's algorithm extends to treat problems of multiple definite summation and integration. That is in general we are interested to determine the annihilating ideal for

$$\sum_{x_1=a_1}^{b_1} \cdots \sum_{x_k=a_k} \int_{x_{k+1}=a_{k+1}}^{b_{k+1}} \cdots \int_{x_m=a_m}^{b_m} f(x_1,\ldots,x_m,Y)\mathrm{d}(x_{k+1},\ldots,x_m), \tag{3.16}$$

where f is a holonomic sequence in the summation variables $x_1,\ldots,x_k$, a holonomic functions in the integration variables $x_{k+1},\ldots,x_m$ and $Y=(y_1,\ldots,y_n)$ is a vector of free parameters. The presence of free parameters is essential for the method to work. If the given multisum or -integral evaluates to a number, then it is not applicable.

Creative telescoping for a generic input such as (3.16) follows the same idea as introduced earlier for Zeilberger's algorithm. Given the annihilating ideal $I = \mathrm{ann}(f)$ (that can be found, e.g., using Koutschan's package) we search the ideal for an operator that acts only on the free parameters, the *principal part*, and for a *delta part*. For discrete variables k this delta part contains as factor the forward difference $\Delta_k = S_k - 1$, for a continuous variable x it contains as factor the derivation with respect to x.

Let us illustrate this on a simple example with two continuous variables x, y and two discrete variables k, n, where we consider y and n as the free parameters, i.e., we want to find the annihilating ideal for

$$F_n(y) = \sum_{k=k_0}^{k_1} \int_{x=x_0}^{x_1} f_{k,n}(x,y)\,\mathrm{dx}$$

in the ring $A = \mathbb{Q}(x, y, k, n)[D_x, D_y, S_k, S_n]$. Given $\operatorname{ann}(f)$, we look for operators $P \in \mathbb{Q}(y, n)[D_y, S_n]$ and $Q \in A$ such that

$$P + \Delta_k D_x Q \in I.$$

If we apply this operator to the given holonomic object $f_{k,n}(x, y)$ this gives

$$P \bullet f_{k,n}(x, y) + \Delta_k D_x Q \bullet f_{k,n}(x, y) = 0.$$

Note that since P does not depend on x it commutes with the integration. Next we integrate the above equation with respect to x:

$$P \bullet \int_{x=x_0}^{x_1} f_{k,n}(x, y)\,\mathrm{dx} + \Delta_k \int_{x=x_0}^{x_1} D_x Q \bullet f_{k,n}(x, y)\,\mathrm{dx} = 0.$$

The integral over the delta part is easily evaluated by the fundamental theorem of calculus and we continue

$$P \bullet \int_{x=x_0}^{x_1} f_{k,n}(x, y)\,\mathrm{dx} + \Delta_k \left(Q \bullet f_{k,n}(x_1, y) - Q \bullet f_{k,n}(x_0, y)\right) = 0.$$

In the next step we sum over k. Then the operator P again commutes with this action and we are left with P acting on $F_n(y)$. The sum over the delta part can be evaluated by telescoping and in short we denote the result as $r_n(y)$. All in all we obtain

$$P \bullet F_n(y) + r_n(y) = 0, \quad \text{i.e.,} \quad \sum_{i=0}^{I} \sum_{j=0}^{J} a_{ij}(n, y) F_{n+i}^{(j)}(y) + r_n(y) = 0.$$

In many applications the remainder of the delta part $r_n(y)$ vanishes and we obtain a homogeneous recurrence rightaway. Otherwise we can determine an annihilating operator $R \in \mathbb{Q}(n, y)[S_n, D_y]$ for r and apply it from the left. The resulting operator $R \circ P$ still annihilates $F_n(y)$ and gives a homogeneous recurrence.

Note also that usually the summation and integration are assumed to run over all the integers and reals, respectively. If the given holonomic function has only finite support, then $r_n(y) = 0$ which is commonly referred to as natural boundary conditions. Note that the sum discussed in example 3.18 has natural boundary conditions because of the presence of the binomial coefficient $\binom{n}{k}$. The algorithm that Annihilator internally used in deriving a recurrence or mixed relation for this sum in fact is just creative telescoping.

Example 3.20. *It is possible to convert expansions in one polynomial bases to another. In some cases the connecting coefficients can be expressed in simple closed form, but at least recurrence relations for these coefficients might be found. If we want to convert a polynomial (or power series) given in its monomial expansion into its Legendre polynomial expansion, then we need to compute the coefficients $a_{k,n}$ in*

$$x^n = \sum_{k=0}^{n} \frac{2}{2k+1} a_{k,n} P_k(x).$$

The factor $h_k = \frac{2}{2k+1}$ enters because of the normalization of the L^2-norm of $P_k(x)$. These Fourier coefficients are then computed as

$$a_{k,n} = \int_{-1}^{1} x^n P_k(x)\,\mathrm{dx}.$$

The integrand above is certainly holonomic, so we can use creative telescoping:

In[24]:= **ann = Annihilator[Integrate[x^kLegendreP[n, x], {x, −1, 1}], {S[k], S[n]}, Assumptions → Element[k, Integers]&&Element[n, Integers]]**

Out[24]= $\{(k-n+1)S_k+(-k-1)S_n, (-k-n-3)S_n^2+(k-n)\}$

We need to add the assumptions on n, k being integers in the command, because otherwise HolonomicFunctions cannot evaluate the delta part at the boundaries of the integral correctly. If we want to see the output in a more traditional form we do the following

In[25]:= **ApplyOreOperator[ann, a[k, n]]**

Out[25]= $\{(-1-k)a[k,1+n]+(1+k-n)a[1+k,n], (k-n)a[k,n]+(-3-k-n)a[k,2+n]\}$

3.6 SumCracker

The method that we discuss next has been developed and implemented by Manuel Kauers [10, 9]. The input is more general than for the previously discussed packages. SumCracker can deal with sequences that are described by systems of difference equations that may be coupled, nonlinear or of higher order with rational function coefficients. They need to be *admissible* in the sense that the system can be used to generate the defined sequences.

The scope of the package includes deciding zero equivalence, finding algebraic dependencies and proving inequalities. The latter is not within the scope of this lecture, but we will introduce the first two. The application of the package is restricted to univariate sequences and if sums appear to indefinite sums only. Also it only deals with shifts, not derivatives.

There is a certain overlap in scope of SumCracker with the packages discussed previously as, e.g., HolonomicFunctions, but mostly they should be applied to different classes of problems as we will show also below. The algorithms implemented in SumCracker also use Gröbner basis computations, but not in a non-commutative setting with operators involved, but we are going back to the classiscal polynomial setting.

First we discuss the algorithm for *deciding zero equivalence*. The task is given sequences $a_1(n), \dots, a_d(n)$ in terms of their defining recurrence relations (or system of), prove polynomial identities

$$f(n) = p(a_1(n), \dots, a_d(n)) = 0, \quad \forall\, n \geq 0,$$

for some $p \in K[x_1, \dots, x_d]$. For instance, prove identity (2.8),

$$f(n) = T_{n+2}(x) - xT_{n+1}(x) + (1-x^2)U_n(x) = 0 \tag{3.17}$$

on Chebyshev polynomials of the first and second kind. For this example we have, e.g.,

$$a_1(n) = T_{n+2}(x), \quad a_2(n) = T_{n+1}(x) \quad \text{and} \quad a_3(n) = U_n(x),$$

and the polynomial p is linear. Note that this is a proper input for the algorithm, since both Chebyshev polynomials satisfy three term recurrences. This means in particular that they are admissible and we can generate the sequence given initial values.

The basic idea is to prove the given statement by induction. In the first step we want to determine a number $N > 0$ such that

$$\forall\, n \geq 0 : f(n) = f(n+1) = \cdots = f(n+N) = 0 \quad \Rightarrow \quad f(n+N+1) = 0. \tag{3.18}$$

Then in the second step it remains to check initial values and we are done. But how can we compute N? For this we reformulate the given identity as a purely polynomial statement.For the example (3.17) above we do so by introducing variables t_0, t_1, u_0, u_1 that correspond to Chebyshev polynomials of first and second kind $T_n, T_{n+1}, U_n, U_{n+1}$. By means of the Chebyshev three term recurrences all higher shifts of Chebyshev polynomials can be expressed solely by t_0, t_1 and x (or u_0, u_1 and x, respectively):

$$\begin{aligned} T_n(x) &\equiv t_0 \\ T_{n+1}(x) &\equiv t_1 \\ T_{n+2}(x) &\equiv 2xt_1 - t_0 \\ T_{n+3}(x) &\equiv (4x^2-1)t_1 - 2xt_0 \\ &\cdots\cdots \end{aligned}$$

And we can proceed analogously for Chebyshev polynomials of the second kind. Note that also the discrete variable n, if it appears in the input has to be translated to a continuous variable and it also exhibits a certain shift behaviour. Now it is also obvious why it is necessary to deal with admissible sequences. In order to carry out the above "translation procedure", we need to be able to express the highest order terms by lower order ones. The result of this transformation turns the given sequence $f(n)$ into a polynomial f_0 given by

$$f(n) \equiv f_0 = xt_1 - t_0 + (1-x^2)u_0.$$

If we denote the shifted versions $f(n+m)$ by $f_m = f_m(t_0, t_1, u_0, u_1, x)$, then the induction proof reads in these terms as

$$f_0 = 0 \wedge f_1 = 0 \wedge f_2 = 0 \wedge \cdots \wedge f_N = 0 \Longrightarrow f_{N+1} = 0. \tag{3.19}$$

On the left hand side we have a set of polynomials generating an ideal

$$I = \langle f_0, f_1, f_2, \ldots, f_N \rangle.$$

Deciding whether (3.19) holds corresponds to deciding if f_{N+1} is in the radical ideal $\mathrm{Rad}(I)$ and this can be done using Gröbner bases computations. The process starts with a certain length of the induction hypothesis depending on the length of the recurrences of the given sequences. If the implication (3.18) does not hold, then the initial value is checked and (if it zero), the length of the induction hypothesis is increased. Kauers showed in his thesis [9] that the algorithm is correct and *terminates*. The algorithm has been implemented in the package SumCracker and the corresponding command is "ZeroSequenceQ".

```
In[26]:= ZeroSequenceQ[ChebyshevT[n + 2, x] − xChebyshevT[n + 1, x] + (1 − x^2)ChebyshevU[n, x]]

Out[26]= True
```

More interesting than proving certain identities may be finding algebraic dependencies of given sequences. Note that (linear) recurrence relations are also a type of algebraic relation. Given a set of admissible sequences, say $(a_n)_{n\geq 0}, (b_n)_{n\geq 0}, (c_n)_{n\geq 0}$, we define the *annihilating ideal* of these sequences as

$$\mathrm{ann}(a_n, b_n, c_n) = \{p \in K[x,y,z] \mid p(a_n, b_n, c_n) \geq 0,\ \forall\, n \geq 0\}.$$

Example 3.21. *Let F_n denote the Fibonacci numbers defined via the recurrence $F_{n+2} = F_{n+1} + F_n$ with $F_0 = 0, F_1 = 1$. Then with the sequences*

$$a_n = F_n, \quad b_n = F_{n+1} \quad \textit{and} \quad c_n = (-1)^n,$$

then Cassini's identity gives rise to an element in the annihilating ideal of these sequences:

$$F_{n-1}F_{n+1} - F_n^2 - (-1)^n = 0 \quad \longleftrightarrow \quad p(x,y,z) = x(x+y) - y^2 - z \in \operatorname{ann}(a_n, b_n, c_n).$$

Note that as indicated in the example above we again do not work with the sequences as such but with their continuous counterparts in some suitable difference ideal. The annihilating ideal is finitely generated, because it is an ideal in a ring of polynomials with finitely many variables. But usually we do not know (and cannot determine) its dimension. Hence we cannot compute the annihilating ideal in general, but it can be approximated from below.

For this first we fix a total degree d. Then we built an ansatz with polynomials up to degree d, e.g., for $d = 2$ and two sequences (i.e., variables),

$$p(x,y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2.$$

Since the sequences we are dealing with are given by a system of difference equations, we can evaluate p for specific instances and solve for the coefficients a_i that are just numbers. Recall once more that also discrete variables like, e.g., n are sent to continuous variables where we model the shift behaviour accordingly.

If a candidate for an identity (algebraic relation) has been found, the zero equivalence decision procedure is used to discard wrong identities. Then the total degree d is increased and the procedure is repeated until some (arbitrary) degree bound d_{max} is reached. Note that *no* upper bound for the degree can be computed. Thus after the execution of the algorithm, we know that all algebraic relations up to degree d_{max} have been found. In the general case, it cannot be determined how many relations we miss (i.e., if we have found all).

The corresponding command in SumCracker is "ApproximateAnnihilator", where *approximate* indicates that not a description of the full annihilating ideal is computed. As an example we derive automatically Cassini's identity for Fibonacci numbers:

In[27]:= $\mathbf{ApproximateAnnihilator[\{Fibonacci[}\boldsymbol{n}\mathbf{], Fibonacci[}\boldsymbol{n}\mathbf{+1], (-1)^{}}\boldsymbol{^n}\mathbf{\}]}$

Out[27]= $\{(-1)^{2n} - 1, (F_n)^2 + F_{n+1}F_n - (F_{n+1})^2 + (-1)^n\}$

Variants of ApproximateAnnihilator that in fact are special cases of this command are "GetLinearRecurrence" and "Crack". GetLinearRecurrence finds (if possible) a linear recurrence for the given sequence:

In[28]:= $\mathbf{GetLinearRecurrence[HermiteH[2}\boldsymbol{n}\mathbf{,}\boldsymbol{x}\mathbf{], In \to }\boldsymbol{n}\mathbf{, Head \to }\boldsymbol{h}\mathbf{]}$

Out[28]= $h[2+n] == -8(1+n)(1+2n)h[n] - 2(5+4n-2x^2)h[1+n]$

The option "In" is used to specify which is the discrete variable that the recurrence is to be determined in. The option "Head" specifies which variable to use for the resulting sequence - the default value is "SUM". SUM is also what is used in SumCracker to denote sums in order to distinguish from the Mathematica built-in "Sum"-command.

In[29]:= $\mathbf{GetLinearRecurrence[SUM[(2}\boldsymbol{k}\mathbf{-1)JacobiP[}\boldsymbol{k}\mathbf{,1,0,}\boldsymbol{x}\mathbf{], \{}\boldsymbol{k}\mathbf{,0,}\boldsymbol{n}\mathbf{\}], In \to }\boldsymbol{n}\mathbf{, Head \to }\boldsymbol{s}\mathbf{]}$

Out[29]= $s[n+3] == \frac{4n^2x+2n^2+24nx+11n+35x+13}{(n+4)(2n+3)}s[n+2]$
$- \frac{8n^3x+4n^3+52n^2x+28n^2+94nx+63n+35x+43}{(n+4)(2n+1)(2n+3)}s[n+1] + \frac{(n+2)(2n+7)}{(n+4)(2n+1)}s[n]$

Note that SumCracker handles only *indefinite* sums, i.e., the upper summation bound is not allowed to appear in the summand. The "Crack" command tries to express a given sum in terms of the objects (sequences) that appear in the summand:

In[30]:= **Crack[SUM[(2*k* + 1)LegendreP[*k*, *x*], {*k*, 0, 2*n*}]]**

Out[30]= $-\frac{(2n+1)(P_{2n}(x) - P_{2n+1}(x))}{x-1}$

In this example above the summand consists of Legendre polynomials, n and the continuous variable x. These are exactly the parts that the closed form expression is constructed from. If for some reason we want to translate the given expression into a different form then we can specify this using the "Into"-option:

In[31]:= **Crack[SUM[(2*k* + 1)LegendreP[*k*, *x*], {*k*, 0, 2*n*}], Into → {JacobiP[2*n*, 1, 0, *x*], *n*}]**

Out[31]= $(1+2n)$JacobiP$[2n, 1, 0, x]$

Chapter 4

Some further applications

In the final section we give some further examples for applications of the symbolic methods presented in the previous section. These examples are partly reviewing some of the classical results presented in section 2 and partly we discuss further questions arising when dealing with orthogonal polynomials.

In Theorem 2.9 several properties of Legendre polynomials were stated that ultimately led to derive the Legendre three term recurrence and to the Legendre differential equation. All these relations are special instances of identities that hold for all Jacobi polynomials.

First note that Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$ are holonomic in the continuous variable x and in the discrete variables α, β and n. A basis for the annihilating ideal in the corresponding operators can be computed using HolonomicFunctions:

In[32]:= **jacann = Factor[Annihilator[JacobiP[n, α, β, x], {$S[n], S[a], S[b], Der[x]$}]]**

Out[32]= $\{(n+\alpha+\beta+1)S_\beta + (1-x)D_x + (-n-\alpha-\beta-1), (n+\alpha+\beta+1)S_\alpha + (-x-1)D_x + (-n-\alpha-\beta-1), 2(n+1)(n+\alpha+\beta+1)S_n - (x-1)(x+1)(2n+\alpha+\beta+2)D_x - (n+\alpha+\beta+1)(2nx + x\alpha + x\beta + 2x + \alpha - \beta), (x-1)(x+1)D_x^2 + (x\alpha + x\beta + 2x + \alpha - \beta)D_x - n(n+\alpha+\beta+1)\}$

Next recall the identity (2.14)

$$(2n+1)P_n(x) = P'_{n+1}(x) - P'_{n-1}(x)$$

stated in theorem 2.9. The corresponding identity for Jacobi polynomials can be found given the basis for the annihilating ideal `jacann` using the "FindRelation"-command of Holnomic-Functions. In HolonomicFunctions we only deal with forward shifts hence we shift (2.14) by one

$$(2n+2)P_{n+1}(x) = P'_{n+2}(x) - P'_n(x).$$

The support of this relation in operator notation is $\{S_n, S_n^2 D_x, D_x\}$. Next we check if there is an immediate generalization of this form available for Jacobi polynomials:

In[33]:= **FindRelation[jacann, Support → {$S[n], S[n]^2 Der[x], Der[x]$}]**

Out[33]= $\{2(n+\alpha+\beta+1)(n+\alpha+\beta+2)(2nx + x\alpha + x\beta + 2x + \alpha - \beta)S_n^2 D_x - (n+\alpha+\beta+1)(2n+\alpha+\beta+3)\left(4n^2x + 4nx\alpha + 4nx\beta + 12nx + x\alpha^2 + 2x\alpha\beta + 6x\alpha + x\beta^2 + 6x\beta + 8x + \alpha^2 + 2\alpha - \beta^2 - 2\beta\right)S_n - 2(n+\alpha+1)(n+\beta+1)(2nx + x\alpha + x\beta + 4x - \alpha + \beta)D_x\}$

If we spell out this result in traditional notation then we obtain

$$\begin{aligned}&2(n+\alpha+\beta+2)\left(\alpha-\beta+x(2+\alpha+\beta+2n)\right)\frac{d}{dx}P_{n+2}^{(\alpha,\beta)}(x)=\\&(2n+\alpha+\beta+3)\left((\alpha-\beta)(\alpha+\beta+2)+x(2n+\alpha+\beta+2)(2n+\alpha+\beta+4)\right)P_{n+1}^{(\alpha,\beta)}(x)\\&+\frac{2(n+\alpha+1)(n+\beta+1)}{n+\alpha+\beta+1}\left(\beta-\alpha+x(2n+\alpha+\beta+4)\right)\frac{d}{dx}P_n^{(\alpha,\beta)}(x).\end{aligned}\tag{4.1}$$

Note that in this relation the appearing coefficients depend on x which was not the case in the corresponding identity for Legendre polynomials. Recall that Legendre polynomials belong to a special subclass of Jacobi polynomials namely the ultraspherical or Gegenbauer polynomials. These are Jacobi polynomials with $\alpha=\beta$. In (4.1) we see that for this choice the identity already would simplify quite a bit. Concerning a generalization of (2.14) to an identity valid for all Jacobi polynomials we should extend the prescribed support to include also S_nD_x. As a rule of thumb we may assume that this central term gets reduced to zero if we specialize to Gegenbauer polynomials. Hence we call FindRelation again with an extended support and with the restriction that the coefficients shall be independent of x:

In[34]:= **FindRelation[jacann, Support → {*S*[*n*], *S*[*n*]²*Der*[*x*], *Der*[*x*], *S*[*n*]*Der*[*x*]}, Eliminate → {*x*}]**

Out[34]= $\{2(n+\alpha+\beta+1)(n+\alpha+\beta+2)(2n+\alpha+\beta+2)S_n^2D_x+2(\alpha-\beta)(n+\alpha+\beta+1)(2n+\alpha+\beta+3)S_nD_x$
$-(n+\alpha+\beta+1)(2n+\alpha+\beta+2)(2n+\alpha+\beta+3)(2n+\alpha+\beta+4)S_n-2(n+\alpha+1)(n+\beta+1)(2n+\alpha+\beta+4)D_x\}$

This identity is now a full generalization of the mixed relation for Legendre polynomials in both support and shape. We also see in this relation the coefficient $\alpha-\beta$ appearing for the term S_nD_x that vanishes for Legendre and in general for Gegenbauer polynomials.

It is also possible to express Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$ in terms of $P_n^{(\alpha+1,\beta+1)}(x)$, i.e., to do a basis transformation. In order to obtain the corresponding connection coefficients we proceed similarly as before:

In[35]:= **FindRelation[jacann, Support → {*S*[*a*]*S*[*b*], *S*[*a*]*S*[*b*]*S*[*n*], *S*[*a*]*S*[*b*]*S*[*n*]², *S*[*n*]²},
Eliminate → {*x*}]**

Out[35]= $\{-(n+\alpha+\beta+3)(n+\alpha+\beta+4)(2n+\alpha+\beta+4)S_n^2S_\alpha S_\beta-(\alpha-\beta)(n+\alpha+\beta+3)(2n+\alpha+\beta+5)S_nS_\alpha S_\beta+$
$(2n+\alpha+\beta+4)(2n+\alpha+\beta+5)(2n+\alpha+\beta+6)S_n^2+(n+\alpha+2)(n+\beta+2)(2n+\alpha+\beta+6)S_\alpha S_\beta\}$

Again this identity can be spelled out in traditional notation and then reads as

$$\begin{aligned}P_{n+2}^{(\alpha,\beta)}(x)=&-\frac{(\alpha+n+2)(\beta+n+2)}{(\alpha+\beta+2n+4)(\alpha+\beta+2n+5)}P_n^{(\alpha+1,\beta+1)}(x)\\&+\frac{(\alpha-\beta)(\alpha+\beta+n+3)}{(\alpha+\beta+2n+4)(\alpha+\beta+2n+6)}P_{n+1}^{(\alpha+1,\beta+1)}(x)\\&+\frac{(\alpha+\beta+n+3)(\alpha+\beta+n+4)}{(\alpha+\beta+2n+5)(\alpha+\beta+2n+6)}P_{n+2}^{(\alpha+1,\beta+1)}(x).\end{aligned}\tag{4.2}$$

Note that also here the identity simplifies if we consider Gegenbauer polynomials. The connection coefficients for Jacobi polynomials $P_n^{(\alpha,\beta)}(x)$ with Jacobi polynomials with parameters α and β shifted forward have the simple, fixed finite support as shown in (4.2). These connection coefficients can also be computed in the general case for transferring $P_n^{(\alpha,\beta)}(x)$ to $P_n^{(\gamma,\delta)}(x)$, however the support usually is running over the whole history $k=0,\dots,n$.

In the context of Chebyshev polynomials we discussed the linearization of products of orthogonal polynomials, i.e., to determine the coefficients $a(k, m, n)$ in

$$\phi_n(x)\phi_m(x) = \sum_{k=0}^{m+n} a(k, m, n)\phi_k(x),$$

for a given orthogonal sequence $(\phi_n(x))_{n\geq 0}$. With $h_n = \int \phi_n(x)^2 w(x)$ dx these coefficients can be computed as the triple product integrals

$$a(k, m, n) = \frac{1}{h_k} \int \phi_n(x)\phi_m(x)\phi_k(x)w(x) \text{ dx}.$$

Let us try to determine these linearization coefficients for $\phi_k(x) = P_k(x)$ and $w(x) \equiv 1$, i.e., for Legendre polynomials. For this purpose we use creative telescoping as implemented in HolonomicFunctions. First we compute the annihilator for the integrand including the inverse squared L^2-norm of Legendre polynomials $\frac{1}{h_n} = \frac{2n+1}{2}$:

In[36]:= **ann = Factor[Annihilator[LegendreP[*k*, *x*]LegendreP[*m*, *x*]LegendreP[*n*, *x*]$\frac{2k+1}{2}$, {*S*[*k*], *S*[*m*], *S*[*n*], *Der*[*x*]}]];**

This annihilator is used as input to execute creative telescoping for the integration with respect to x:

In[37]:= **conn = Factor[CreativeTelescoping[ann, *Der*[*x*], {*S*[*k*], *S*[*m*], *S*[*n*]}]];**

In[38]:= **Factor[conn[[1]]]**

Out[38]= $\{(k+m-n+1)(k-m+n)S_m - (k+m-n)(k-m+n+1)S_n, (2k+1)(k-m-n)(k+m-n+1)S_k - (2k+3)(k-m-n-1)(k+m-n)S_n, (k-m-n-2)(k+m-n-1)(k-m+n+2)(k+m+n+3)S_n^2 - (k-m-n-1)(k+m-n)(k-m+n+1)(k+m+n+2)\}$

Note that the computations are rather involved and take some time even on big machines. The output of CreativeTelescoping contains both the principal part displayed above and the delta part for which one needs to check that it indeed telescopes to zero. We omit this step here. Finally we are interested in a recurrence relation for the linearization coefficients in k. To find such a recurrence we use the FindRelation command:

In[39]:= **krec = Factor[FindRelation[conn[[1]], Support → {1, *S*[*k*], *S*[*k*]^2}]]**

Out[39]= $\{(2k+1)(k-m-n+1)(k+m-n+2)(k-m+n+2)(k+m+n+3)S_k^2 - (2k+5)(k-m-n)(k+m-n+1)(k-m+n+1)(k+m+n+2)\}$

In[40]:= **ApplyOreOperator[krec, *a*[*k*]]**

Out[40]= $\{-((5+2k)(k-m-n)(1+k+m-n)(1+k-m+n)(2+k+m+n)a[k]) + (1+2*k)(1+k-m-n)(2+k+m-n)(2+k-m+n)(3+k+m+n)a[2+k]\}$

This recurrence can then be solved using the Mathematice built-in RSolve command. Note that the recurrence progresses in steps of size two. This is natural if we recall that Legendre polynomials are even for even degrees and odd for odd degrees. Hence if the product $P_n(x)P_m(x)$ is a polynomial of even degree then all the odd coefficients in the linearization vanish and vice versa. Thus also a common way to write them is as $a(m+n-2k, m, n)$ instead of $a(k, m, n)$. It is also easy to see that then the upper summation bound is $\min(m, n)$ only. We summarize the result in the following corollary following [2, Cor. 6.8.3].

Corollary 4.1. *For Legendre polynomials* $P_n(x)$

$$P_n(x)P_m(x) = \sum_{k=0}^{\min\{m,n\}} \frac{2m+2n+1-4k}{2m+2n+1-2k} \frac{\left(\frac{1}{2}\right)_k \left(\frac{1}{2}\right)_{m-k} \left(\frac{1}{2}\right)_{n-k} (m+n-k)!}{k!(m-k)!(n-k)! \left(\frac{1}{2}\right)_{m+n-k}} P_{m+n-2k}(x).$$

In high order finite element methods integrals over given coefficient functions f with products of (usually) orthogonal polynomials need to be computed. In the simple one dimensional case this means that integrals of the form

$$A_{i,j} = \int_a^b f(x)\phi_i(x)\phi_j(x) \, \mathrm{dx}$$

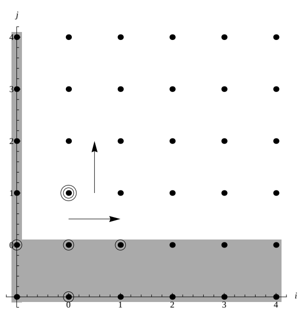
need to be evaluated for polynomial degrees i, j up to an upper bound p giving rise to a system matrix $A = (A_{i,j})_{i,j=0}^p$. If we can determine a recurrence for the product $\phi_i(x)\phi_j(x)$ with shifts in i, j whose coefficients do not depend on x then this recurrence can be used to build up the matrix A. For all orthogonal polynomials it is possible to derive such a recurrence relation based on the respective three term recurrence. For specific sequences $(\phi_n(x))_{n\geq 0}$ this can be carried out automatically using, e.g., HolonomicFunctions or packages with a similar scope. As a simple example consider the product of two Jacobi polynomials $\psi_{ij}(x) = P_i^{(2,0)}(x)P_j^{(1,1)}(x)$. Note that this is not merely product of two polynomials of the same family. An x-free recurrence exists nonetheless however, if both of the given polynomial sequences are orthogonal with respect to some weight function and satisfy a three term recurrence in the standard form.

For the computations first derive an annihilator for the product ψ_{ij} with shifts in i, j and then the annihilating ideal is searched for a relation where no x appears in the coefficients:

```
In[41]:= ann = Factor[Annihilator[JacobiP[i, 2, 0, x]JacobiP[j, 1, 1, x], {S[i], S[j]}]];
In[42]:= rec = Factor[FindRelation[ann, Eliminate → {x}]]
```

Out[42]= $\{(i+2)^2(i+4)(j+3)(2j+5)S_i^2S_j - (i+2)(i+3)(2i+5)(j+2)(j+4)S_iS_j^2 - (2i+5)(j+3)(2j+5)S_iS_j - (i+2)(i+3)(2i+5)(j+2)(j+3)S_i + (i+1)(i+3)^2(j+3)(2j+5)S_j\}$

The support of this recurrence is of diamond shape. In the lattice plot it is indicated by the circled grid points. The double circle corresponds to the new matrix entry $A_{i,j}$ that can be computed from given initial values. The necessary initial values are exactly the grey-shaded areas. Note that the graph is centered at $(-1, -1)$. This simplifies the computation of the needed initial values since for all classical orthogonal polynomials the three term recurrence can be extended to degree -1 by setting $\phi_{-1}(x) = 0$.



Starting from these initial values first the entries $A_{i,1}$ for $i = 1, \ldots, 2p$ can be computed. Note that by the shape of the recurrence we need to compute initial values up to degree $2p$ if we need to fill the matrix up to $A_{p,p}$. This is indicated by the arrow pointing to the right. In the second step we increase j and move one line up and compute $A_{i,2}$ for $i = 1, \ldots, 2p - 1$

recursively. This procedure is repeated until the full matrix is computed. If we consider the matrix A built from solely the product of the two orthgonal polynomials (with a constant coefficient function at most), then the initial values are easily computed. In this case a simple routine for building up the matrix is as follows:

In[43]:= $\boldsymbol{A[-1, j\_Integer] := 0;\ A[i\_Integer, -1] := 0;\ A[i\_Integer, 0] := 2;}$

In[44]:= $$\boldsymbol{A[i\_Integer, j\_Integer] := A[i,j] = \frac{i(i+2)(j+1)(2j+1)}{(i+1)(2i+3)j(j+2)}A[i-1, j-1] - \frac{j+1}{j+2}A[i, j-2] - \frac{(j+1)(2j+1)}{(i+1)(i+2)j(j+2)}A[i, j-1] + \frac{(i+1)(i+3)(j+1)(2j+1)}{(i+2)(2i+3)j(j+2)}A[i+1, j-1];}$$

Note that above we used caching in order to speed up the computations and that the initial values $A_{i,0}$ for $i \geq 1$ are just the constant value 2. Then the matrix can be assembled in a fast way:

In[45]:= $\boldsymbol{\text{Timing}[\text{mat} = \text{Table}[A[i,j], \{i, 0, 50\}, \{j, 0, 50\}];]}$

Out[45]= $\{0.200013, \text{Null}\}$

If the coefficient function is not constant, then the recursion can essentially be used in the same way. Only the initial values $A_{i,0}$ need to be adjusted.

Bibliography

[1] M. Abramowitz and I. Stegun, editors. *Handbook of mathematical functions.* Dover-Publications, 1965.

[2] G.E. Andrews, R. Askey, and R. Roy. *Special Functions.* Encyclopedia of Mathematics and its Applications 71. Cambridge University Press, 2000.

[3] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal).* PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965. English translation in J. of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions. Vol. 41, Number 3-4, Pages 475–511, 2006.

[4] T.S. Chihara. *An introduction to orthogonal polynomials.* Gordon and Breach Science Publishers, New York, 1978. Mathematics and its Applications, Vol. 13.

[5] D.J. Dickinson, H.O. Pollak, and G.H. Wannier. On a class of polynomials orthogonal over a denumerable set. *Pacific J. Math.*, 6:239–247, 1956.

[6] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.

[7] J. Favard. Sur les polynomes de Tchebicheff. *Comptes Rendus Acad. Sci. Paris*, 200:2052–2053, 1935.

[8] R.W. Gosper. Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA*, 75:40–42, 1978.

[9] M. Kauers. *Algorithms for Nonlinear Higher Order Difference Equations.* PhD thesis, RISC-Linz, October 2005.

[10] M. Kauers. SumCracker – A Package for Manipulating Symbolic Sums and Related Objects. *Journal of Symbolic Computation*, 41(9):1039–1057, 2006. Available at http://www.risc.uni-linz.ac.at/publications/.

[11] M. Kauers and P. Paule. *The Concrete Tetrahedron.* Text and Monographs in Symbolic Computation. Springer Wien, 1st edition, 2011.

[12] C. Koutschan. *Advanced Applications of the Holonomic Systems Approach.* PhD thesis, RISC-Linz, Johannes Kepler University, September 2009.

[13] C. Koutschan. HolonomicFunctions (User's Guide). Technical Report 10-01, RISC Report Series, University of Linz, Austria, January 2010.

[14] H.S. Wilf M. Petkovšek and D. Zeilberger. $A = B$. A K Peters Ltd., Wellesley, MA, 1996.

[15] C. Mallinger. Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences. Master's thesis, RISC, J. Kepler University, August 1996. Available at http://www.risc.uni-linz.ac.at/publications/.

[16] P. Paule and M. Schorn. A Mathematica Version of Zeilberger's Algorithm for Proving Binomial Coefficient Identities. *Journal of Symbolic Computation*, 20:673–698, 1995.

[17] E.D. Rainville. *Special Functions*. Chelsea Publishing Co., Bronx, N.Y., first edition, 1971.

[18] B. Salvy and P. Zimmermann. Gfun: A package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software*, 20:163–177, 1994.

[19] G. Szegö. *Orthogonal Polynomials*. AMS Colloquium Publications, Volume XXIII. 3 edition, 1974.

[20] P. Verbaeten. The automatic construction of pure recurrence relations. In *Proceedings of Eurosam 74, Sigsam Bulletin*, volume 8, pages 96–98, 1974.

[21] P. Verbaeten. *Rekursiebetrekkingen voor lineaire hypergeometrische funkties*. PhD thesis, Department of Computer Science, K.U.Leuven, Leuven, Belgium, March 1976.

[22] H.S. Wilf. *generatingfunctionology*. A K Peters Ltd., Wellesley, MA, third edition, 2006.

[23] H.S. Wilf and D. Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Invent. Math.*, 108(3):575–633, 1992.

[24] F. Winkler. *Polynomial Algorithms in Computer Algebra*. Texts and Monographs in Symbolic Computation. Springer Wien New York, 1 edition, 1996.

[25] D. Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete. Math.*, 80:207–211, 1990.

[26] D. Zeilberger. A holonomic systems approach to special functions identities. *J. Comput. Appl. Math.*, 32(3):321–368, 1990.