INSTYTUT INFORMATYKI
UNIWERSYTETU WROCŁAWSKIEGO

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW

ul. Przesmyckiego 20
51–151 Wrocław
Poland

Stanisław Lewanowicz and Paweł Woźny

Algorithms for construction of recurrence relations for the coefficients of expansions in series of classical orthogonal polynomials

February 2001

ALGORITHMS FOR CONSTRUCTION OF RECURRENCE RELATIONS FOR THE COEFFICIENTS OF EXPANSIONS IN SERIES OF CLASSICAL ORTHOGONAL POLYNOMIALS (*)

STANISLAW LEWANOWICZ AND PAWEŁ WOŹNY

Abstract. Let $\{P_k(x)\}$ be any sequence of classical orthogonal polynomials, i.e., (i) very classical orthogonal polynomials, or (ii) classical orthogonal polynomials of a discrete variable, or (iii) q-classical orthogonal polynomials in the Hahn sense.

Maple implementation is given of an algorithmic approach to the problem of constructing recurrence relations for the coefficients of the Fourier series expansions with respect to the polynomials $\{P_k(x)\}$.

1. Introduction

Let $\{P_k(x)\}$ be any sequence of classical orthogonal polynomials, i.e.,

- very classical orthogonal polynomials (VCOP, in short) [8, pp. 150–152], or
- classical orthogonal polynomials of a discrete variable (DCOP, in short) [34, Chapter II], or
- q-classical orthogonal polynomials in the Hahn sense (qCOP, in short) [13, 16, 31, 32].

Given a function f, expansion in series

$$f = \sum_k a_k[f] P_k \tag{1.1}$$

is a matter of interest in numerical analysis, applied mathematics and mathematical physics. Important special cases are *connection problem*,

$$\bar{P}_n(x) = \sum_k c_k P_k \qquad (n \in \mathbb{Z}^+),$$

and *(generalized) linearization problem*,

$$\bar{P}_m(x)\bar{P}_n(x) = \sum_k l_k P_k \qquad (m,\, n \in \mathbb{Z}^+),$$

where $\{\bar{P}_k(x)\}$ is a family of polynomials (e.g., belonging to the same class as $\{P_k(x)\}$). In particular, positivity of the connection coefficients c_k, or linearization coefficients l_k is of great importance. See [1, 2, 3, 4, 5, 10, 21, 22, 23, 27, 29, 44, 45, 46].

Usually, the determination of the expansion coefficients $a_k[f]$ requires a deep knowledge of special functions (from hypergeometric, or basic hypergeometric families). It is important to remark that, even in the case when it is possible to compute explicitly these coefficients, it is often useful to have a recurrence relation of the type

$$\sum_{i=0}^{r} A_i(k)\, a_{k+i}[f] = B(k), \tag{1.2}$$

where $r \in \mathbb{N}$, and A_i and B are known functions of k. Equation (1.2) may serve as a tool for detection certain properties of $a_k[f]$, or for numerical evaluation of these quantities.

(*) Supported by Komitet Badań Naukowych (Poland) under the grant 2P03A02717.
Corrected and updated version. May 20, 2003.

In the present paper, we give Maple algorithms implementing the methods described in [26, 28, 48], which, under assumption that the function $f(x)$ satisfies a linear differential equation (VCOP case), or linear difference (q-difference) equation (DCOP and qCOP cases) with polynomial coefficients, construct the recurrence relations for the coefficients $a_k[f]$ of the form (1.2), where the functions A_i are given in terms of the polynomial coefficients σ and τ of the Pearson-type q-difference equation for the orthogonality weight ϱ associated with $\{P_k(x)\}$, and the coefficients of the three-term recurrence relation and of the structure relations satisfied by the polynomials $\{P_k(x)\}$.

2. Very classical polynomials

2.1. Properties of the very classical orthogonal polynomials. Let $P_n^{(\alpha,\beta)}(x)$, $L_n^{\alpha}(x)$, $Y_n^{\alpha}(x)$ and $H_n(x)$ denote the monic *Jacobi*, *Laguerre*, *Bessel* and *Hermite polynomials* (see Appendix A, Table 1). These families are members of the class of the very classical orthogonal polynomials [8, pp. 150–152] (VCOP, in short).

Any sequence $P_k = P_k(x)$ $(k \geq 0)$ from the class VCOP, has the orthogonality property,

$$\int_A^B \varrho P_k P_l \, dx = h_k \delta_{kl}, \qquad h_k \neq 0, \tag{2.1}$$

where ϱ is the weight function, which satisfies the *Pearson equation*

$$\mathbf{D}(\sigma\varrho) = \tau\varrho$$

with $\mathbf{D} := \frac{d}{dx}$, where σ and τ are polynomials, $\deg\sigma \leq 2$, $\deg\tau = 1$,

Besides three-term recurrence relation,

$$xP_k(x) = \mathcal{X}P_k(x), \text{ where } \mathcal{X} := \xi_0(k)\,\mathcal{E}^{-1} + \xi_1(k)\mathcal{I} + \xi_2(k)\,\mathcal{E},$$

any family $\{P_k\}$ belonging to this class, has the following properties [8, pp. 150–152]; [15], [33, Ch. II]. First, for $k \geq 0$ the second-order differential equation

$$\mathbf{L}_k P_k \equiv \{\sigma\mathbf{D}^2 + \tau\mathbf{D} + \lambda_k\mathbf{I}\}P_k = 0$$

holds with

$$\lambda_k := k\left\{(1-k)\sigma''/2 - \tau'\right\}.$$

Second, we have the *structure relation*

$$\sigma\mathbf{D}P_k = \lambda_k\mathcal{D}P_k, \text{ where } \mathcal{D} := \delta_0(k)\,\mathcal{E}^{-1} + \delta_1(k)\mathcal{I} + \delta_2(k)\,\mathcal{E}.$$

Third, for any root ζ of σ, we have the differential-recurrence identity,

$$\frac{\sigma(x)}{x-\zeta}\mathcal{Q}\left(\lambda_k^{-1}\mathbf{D}P_k(x)\right) = \mathcal{P}_\zeta P_k(x),$$
$$\text{where } \mathcal{P}_\zeta := \mathcal{I} + \pi(\zeta;k)\,\mathcal{E}, \quad \mathcal{Q} := \vartheta(k)\,\mathcal{I} + \omega(\zeta;k)\,\mathcal{E},$$

where

$$\vartheta(k) := \xi_0(k)/\delta_2(k).$$

Here $\mathcal{I}$ is the identity operator, $\mathcal{I}z_k = z_k$, and $\mathcal{E}^m$ is the mth shift operator, $\mathcal{E}^m z_k = z_{k+m}$ $(m \in \mathbb{Z})$. For the sake of simplicity, we write $\mathcal{E}$ in place of $\mathcal{E}^1$. (We adopt the convention that all the script letter operators act on the variable k).

The *Fourier coefficients* are defined by

$$a_k[f] := \frac{1}{h_k}\int_A^B \varrho p_k f.$$

2.2. **Basic operators.** Let $\{P_k\}$ be Jacobi polynomials $P_k^{(\alpha,\beta)}$, or Bessel polynomials $Y_n^\alpha(x)$ (see Table 1, in the Appendix A). Given a zero ζ of the polynomial σ associated to $\{P_k\}$, we define the following sequences of operators:

$$\begin{aligned}\mathcal{P}_\zeta^{(m)} &:= \mathcal{I} + \pi_m(\zeta;k)\,\mathcal{E},\\ \mathcal{Q}_\zeta^{(m)} &:= \vartheta(k)\,\mathcal{I} + \omega_m(\zeta;k)\,\mathcal{E}, \qquad\qquad (m = 0,1,\dots)\\ \mathcal{R}_\zeta^{(m)} &:= \delta_0(k)\,\mathcal{E}^{-1} + \rho_m(\zeta;k)\,\mathcal{I},\end{aligned}$$

where

$$\omega_m(\zeta;k) := \frac{\xi_2(k+1)\pi_m(\zeta;k)}{\delta_2(k+1)},$$

and

$$\left.\begin{aligned}\pi_m(\zeta;k) &:= \zeta\frac{(2k+\gamma)_2(2k+\gamma+1)_2}{2(k+\gamma)(k+\nu_{-\zeta})(2k+\gamma+m+1)_2},\\ \rho_m(\zeta;k) &:= -2\zeta\frac{k+\nu_{-\zeta}+m}{(2k+\gamma+m)_2}\end{aligned}\right.\qquad (\zeta\in\{-1,+1\})$$

with $\gamma := \alpha+\beta+1$, and

$$\nu_\zeta := \frac{\zeta\cdot\tau(\zeta)}{\sigma''} = \begin{cases}\beta+1 & (\zeta=-1),\\ \alpha+1 & (\zeta=+1)\end{cases} \tag{2.2}$$

in the Jacobi case, and

$$\left.\begin{aligned}\pi_m(\zeta;k) &:= -\frac{(2k+\alpha+1)_2(2k+\alpha+2)_2}{2(k+\alpha+1)(2k+\alpha+m+2)_2},\\ \rho_m(\zeta;k) &:= \frac{2}{(2k+\alpha+m+1)_2}\end{aligned}\right.\qquad (\zeta=0),$$

in the Bessel case (notice that $\zeta=0$ is the double root of σ in this case).

It can be checked that

$$\begin{aligned}&\mathcal{P}_\zeta^{(0)} = \mathcal{P}_\zeta, \qquad \mathcal{Q}_\zeta^{(0)} = \mathcal{Q}_\zeta, \qquad \mathcal{R}_\zeta^{(0)}\mathcal{P}_\zeta^{(0)} = \mathcal{D},\\ &\mathcal{P}_\zeta^{(m)}\mathcal{Q}_\zeta^{(m-1)} = \mathcal{Q}_\zeta^{(m)}\mathcal{P}_\zeta^{(m-1)} \qquad (m\ge 1),\\ &\mathcal{R}_\zeta^{(m)}\mathcal{P}_\zeta^{(m)} = \mathcal{P}_\zeta^{(m-1)}\mathcal{R}_\zeta^{(m-1)} \qquad (m\ge 1).\end{aligned}$$

Further, for $i,j\ge 0$, let us define

$$\begin{aligned}\mathcal{S}_\zeta^{(i,j)} &:= \begin{cases}\mathcal{I} & (i<j),\\ \mathcal{P}_\zeta^{(i)}\mathcal{P}_\zeta^{(i-1)}\cdots\mathcal{P}_\zeta^{(j)} & (i\ge j),\end{cases}\\ \mathcal{Z}_\zeta^{(i)} &:= \mathcal{S}_\zeta^{(i-1,0)} \qquad (i\ge 0),\\ \mathcal{U}_\zeta^{(i)} &:= \begin{cases}\mathcal{I} & (i=0),\\ \mathcal{Q}_\zeta^{(i-1)}\cdots\mathcal{Q}_\zeta^{(1)}\mathcal{Q}_\zeta^{(0)} & (i\ge 1),\end{cases}\\ \mathcal{M}_\zeta^{(i,j)} &:= \begin{cases}\mathcal{I} & (i>j),\\ \mathcal{R}_\zeta^{(i)}\mathcal{R}_\zeta^{(i+1)}\cdots\mathcal{R}_\zeta^{(j)} & (0\le i\le j),\end{cases}\\ \mathcal{N}_\zeta^{(i)} &:= \mathcal{M}_\zeta^{(0,i-1)}.\end{aligned}$$

2.3. **Algorithms.** Given any sequence $\{P_k\}$ $(k \geq 0)$ of the classical orthogonal polynomials, let us introduce the differential operators $\mathbf{U}$, $\mathbf{L}$ and $\mathbf{Z}_\zeta$ by the following formulae:

$$\begin{aligned}\mathbf{U} &:= \sigma\mathbf{D} + \tau\mathbf{I},\\ \mathbf{L} &:= \mathbf{U}\mathbf{D},\\ \mathbf{Z}_\zeta &:= (x-\zeta)\mathbf{D},\end{aligned}$$

where ζ is any root of σ.

If $P_k = P_k^{(\alpha,\beta)}$ are Jacobi polynomials[(*)], let $\mathbf{K}_\zeta$ $(\zeta = \pm 1)$ be the following special-form differential operator:

$$\mathbf{K}_\zeta := (\mathbf{Z}_\zeta + \nu_\zeta \mathbf{I})\,\mathbf{D},$$

where ν_ζ is given by (2.2).

Let assume that the function $f(x)$ satisfies the following differential equation:

$$\mathbf{P}_n f(x) \equiv \sum_{i=0}^{n} w_n(x)\mathbf{D}^i f(x) = g(x), \tag{2.3}$$

where w_{ni} are polynomials in x, and the function g has known expansion in series of P_k.

The first step of the algorithm is to convert the LHS of the equation (2.3) to the form

$$\mathbf{P}_n f = \sum_{i=0}^{n} \mathbf{Q}_i\,(z_i f), \tag{2.4}$$

where z_i are polynomials, and the differential operators $\mathbf{Q}_i$ have the form

$$\mathbf{Q}_i := \mathbf{D}^{p_i}\mathbf{K}_{\zeta_i}^{q_i}\mathbf{Z}_{\zeta_i}^{r_i}\mathbf{L}^{s_i}\mathbf{U}^{t_i}, \tag{2.5}$$

where ζ_i is a root of the associated polynomial σ, p_i, $s_i \in \mathbb{Z}^+$, $t_i \in \{0,1\}$ and

$$q_i \in \begin{cases} \mathbb{Z}^+ & \text{(Jacobi case)},\\ \{0\} & \text{(Bessel, Laguerre and Hermite cases)},\end{cases} \tag{2.6}$$

$$r_i \in \begin{cases} \{0,1\} & \text{(Jacobi case)},\\ \mathbb{Z}^+ & \text{(Bessel case)},\\ \{0\} & \text{(Laguerre and Hermite case)}.\end{cases} \tag{2.7}$$

To this end, define the differential operators $\mathbf{P}_i$ $(i = 0, 1, \ldots, n-1)$ and $\mathbf{Q}_j$ $(j = 1, 2, \ldots, n)$, and the polynomials $z_0, z_1, \ldots, z_n$ in the following recursive way.

For $i = n,\ n-1, \ldots, 1$, given the operator $\mathbf{P}_i = \sum\limits_{j=0}^{i} w_{ij}\mathbf{D}^j$,

- represent the leading coefficient w_{ii} in the form
$$w_{ii}(x) = [\sigma(x)]^{\alpha_i}(x-\zeta_i)^{\beta_i}u_i(x)$$
where polynomial u_i has no roots in common with σ, and
$$\alpha_i \begin{cases} \in \mathbb{Z}^+ & \text{(Jacobi, Bessel and Laguerre)},\\ = i & \text{(Hermite)},\end{cases}$$
$$\beta_i \begin{cases} \in \mathbb{Z}^+ & \text{(Jacobi and Bessel)},\\ = 0 & \text{(Laguerre and Hermite)},\end{cases}$$
ζ_i being a root of σ (if $\beta_i = 0$, value of ζ_i is inessential), then find out which of the following cases holds:

[(*)] Note that it is the only case where the associate polynomial $\sigma(x)$ has two different real zeros, namely -1 and 1 (cf. Tables 4 and 5).

– case A: $\alpha_i \geq m$,
– case B: $\alpha_i + \beta_i \geq m > \alpha$,
– case C: $m > \alpha_i + \beta_i$,

where $m := \lfloor (i+1)/2 \rfloor$;

- define

$$z_i(x) := u_i(x) \cdot \begin{cases} [\sigma(x)]^{\alpha_i - m}(x-\zeta)^{\beta_i} & \text{(case A)}, \\ (x-\zeta)^{\alpha_i+\beta_i-m} & \text{(case B)}, \\ 1 & \text{(case C)}; \end{cases}$$

- define

$$r_i := \begin{cases} i \mod 2 & \text{(case B)}, \\ 0 & \text{(case A or C)}, \end{cases} \qquad t_i := \begin{cases} i \mod 2 & \text{(case A)}, \\ 0 & \text{(case B or C)}, \end{cases}$$

$$s_i := \min\{\alpha_i,\, \lfloor i/2 \rfloor\} \quad q_i := \min\{\beta_i,\, \lfloor i/2 \rfloor - t_i\}, \quad p_i := i - 2q_i - 2s_i - r_i - t_i;$$

- define the operator $\mathbf{Q}_i$ by

$$\mathbf{Q}_i := \mathbf{D}^{p_i}\mathbf{K}_{\zeta_i}^{q_i}\mathbf{Z}_{\zeta_i}^{r_i}\mathbf{L}^{s_i}\mathbf{U}^{t_i};$$

- define the operator $\mathbf{P}_{i-1}$ of degree $i-1$ as

$$\mathbf{P}_{i-1}f(x) := \mathbf{P}_i f(x) - \mathbf{Q}_i\,(z_i f)(x).$$

For convenience, let us denote

$$\mathbf{Q}_0 := \mathbf{I}, \qquad z_0 := w_{0,0}, \tag{2.8}$$

where $w_{0,0}$ is the only coefficient of the operator $\mathbf{P}_0$.

Algorithm 2.1 ([26]). *Let $\{P_k\}$ be any sequence of the very classical orthogonal polynomials, and let the function f obey the differential equation*

$$\mathbf{P}_n f(x) \equiv \sum_{i=0}^{n} \mathbf{Q}_i\,(z_i f)(x) = g(x), \tag{2.9}$$

where z_i are polynomials, and the ith-order differential operator $\mathbf{Q}_i$ $(i = 0, 1, \ldots, n)$ is of the form

$$\mathbf{Q}_i := \mathbf{D}^{p_i}\mathbf{K}_{\zeta}^{q_i}\mathbf{Z}_{\zeta}^{r_i}\mathbf{L}^{s_i}\mathbf{U}^{t_i}, \tag{2.10}$$

where ζ is a fixed root of the associated polynomial σ, p_i, $s_i \in \mathbb{Z}^+$, $t_i \in \{0,1\}$ and

$$q_i \in \begin{cases} \mathbb{Z}^+ & \textit{(Jacobi case)}, \\ \{0\} & \textit{(Bessel, Laguerre and Hermite cases)}, \end{cases}$$

$$r_i \in \begin{cases} \{0,1\} & \textit{(Jacobi case)}, \\ \mathbb{Z}^+ & \textit{(Bessel case)}, \\ \{0\} & \textit{(Laguerre and Hermite case)}. \end{cases}$$

Define

$$e := \max_{0 \leq i \leq n} p_i, \qquad d := \max_{0 \leq i \leq n} (p_i + 2q_i + r_i) - e,$$

$$\mathcal{T} := \mathcal{Z}_{\zeta}^{(d)}\mathcal{D}^e,$$

and

$$\mathcal{A}_i := \begin{cases} \mu_{q_i}(\zeta;k)\,\mathcal{E}^{q_i}\mathcal{U}_{\zeta}^{(r_i)}\,(-\lambda_k)^{s_i+t_i}\,\mathcal{D}^{t_i} & \textit{(Jacobi and Bessel)}, \\ (-\lambda_k)^{s_i+t_i}\,\mathcal{D}^{t_i} & \textit{(Laguerre and Hermite)}, \end{cases}$$

$$\mathcal{C}_i := \begin{cases} \mathcal{M}_\zeta^{(d,d+e-p_i-1)}\mathcal{S}_\zeta^{(d+e-p_i-1,2q_i+r_i)} & \textit{(Jacobi and Bessel)},\\ \mathcal{D}^{e-p_i} & \textit{(Laguerre and Hermite)}, \end{cases}$$

where $i = 0, 1, \ldots, n$.

Then the Fourier coefficients $a_k[f]$ *satisfy the recurrence relation*

$$\mathcal{L}(h_k a_k[f]) = B(k), \tag{2.11}$$

where

$$\mathcal{L} := \sum_{i=0}^{n} \mathcal{C}_i \mathcal{A}_i z_i(\mathcal{X}), \tag{2.12}$$

$$B(k) := \mathcal{T}(h_k a_k[g]). \tag{2.13}$$

In the Jacobi case, it is possible that not all ζ_i's in (2.5) are equal, so that Algorithm 2.1 is not applicable. The next algorithm is a corrected version of a result in [19].

Algorithm 2.2 ([26]). *Let* $\{P_k\}$ *be the sequence of the Jacobi polynomials, and let* f *satisfies the equation*

$$\mathbf{P}_n f(x) \equiv \sum_{i=0}^{n} \mathbf{Q}_i\,(z_i f)(x) = g(x),$$

where z_i *are polynomials, and the differential operators* $\mathbf{Q}_i$ *have the form*

$$\mathbf{Q}_i := \mathbf{D}^{p_i}\mathbf{K}_{\zeta_i}^{q_i}\mathbf{Z}_{\zeta_i}^{r_i}\mathbf{L}^{s_i}\mathbf{U}^{t_i},$$

where $\zeta_i \in \{-1, +1\}$, $p_i, q_i, s_i \in \mathbb{Z}^+$, $r_i, t_i \in \{0, 1\}$, $p_i+2q_i+r_i+2s_i+t_i = i$, *and the expansion of the function* g *in* P_k *is known.*

Let us introduce the notation

$$\begin{aligned}
&\Omega := \{1, 2, \ldots, n\}, \qquad \Omega_\eta := \{i \in \Omega : \zeta_i = \eta\} \qquad (\eta \in \{-1, +1\}),\\
&v_i := 2q_i + r_i \qquad (i \in \Omega),\\
&e_\eta := \max_{i\in\Omega_\eta} p_i, \qquad d_\eta := \max_{i\in\Omega_\eta}(p_i + v_i) - e_\eta, \qquad (\eta \in \{-1, +1\}),\\
&\omega := \begin{cases} -1 & \textit{if} \quad d_{-1} + e_{-1} \ge d_1 + e_1,\\ +1 & \textit{if} \quad d_{-1} + e_{-1} < d_1 + e_1, \end{cases}\\
&e := \max(e_\omega, d_{-\omega} + e_{-\omega}), \qquad d := d_\omega + e_\omega - e.
\end{aligned}$$

Further, let define difference operators $\mathcal{T}$, $\mathcal{A}_i$, $\mathcal{B}_i$, $\mathcal{C}_i$, $\mathcal{J}_\eta$ *by*

$$\begin{aligned}
&\mathcal{T} := \mathcal{Z}_\omega^{(d)}\mathcal{D}^e,\\
&\mathcal{A}_i := \mu_{q_i}(\omega; k)\,\mathcal{E}^{q_i}\mathcal{U}_\omega^{(r_i)}\,(-\lambda_k)^{s_i+t_i}\,\mathcal{D}^{t_i} \qquad (i = 0, 1, \ldots, n),\\
&\mathcal{B}_i := \mathcal{M}_\eta^{(d_\eta, e_\eta+d_\eta-p_i-1)}\mathcal{S}_\eta^{(e_\eta+d_\eta-p_i-1, v_i)} \qquad (i \in \Omega_\eta;\ \eta = -1, +1),\\
&\mathcal{J}_\omega := \mathcal{M}_\omega^{(d,d_\omega-1)}, \qquad \mathcal{J}_{-\omega} := \mathcal{Z}_\omega^{(d)}\mathcal{N}_{-\omega}^{(e-e_{-\omega})}\mathcal{S}_{-\omega}^{(e-e_{-\omega}-1,d_{-\omega})},\\
&\mathcal{C}_0 := \mathcal{T}, \qquad \mathcal{C}_i := \begin{cases} \mathcal{J}_\omega\mathcal{B}_i & (i \in \Omega_\omega),\\ \mathcal{J}_{-\omega}\mathcal{B}_i & (i \in \Omega_{-\omega}). \end{cases}
\end{aligned} \tag{$*$}$$

The Fourier coefficients $a_k[f]$ *satisfy the recurrence relation*

$$\mathcal{L}(h_k a_k[f]) = B(k),$$

where

$$\mathcal{L} := \sum_{i=0}^{n} \mathcal{C}_i \mathcal{A}_i z_i(\mathcal{X}),$$

$$B(k) := \mathfrak{T}(h_k a_k[g]).$$

3. Discrete classical and q-classical orthogonal polynomials

3.1. Properties of the discrete and q-classical classical orthogonal polynomials. Let $\{P_k\}$ be any member of the family of classical orthogonal polynomials of a discrete variable (DCOP, in short) [34, Chapter II], or q-classical polynomials in the Hahn sense (qCOP, in short) [13, 16, 31, 32], See Appendix A, Tables 2 and 3.

We introduce the notation

$$\mathbf{E}^{\pm} f(x) := f(x \pm 1), \quad \mathbf{D}^{+} := \Delta, \quad \mathbf{D}^{-} := \nabla \qquad \text{(DCOP)},$$
$$\mathbf{E}^{\pm} f(x) := f(xq^{\pm 1}), \quad \mathbf{D}^{+} := \mathbf{D}_q, \quad \mathbf{D}^{-} := \mathbf{D}_{1/q} \qquad \text{(qCOP)}.$$

Here the *forward* and *backward difference operators* Δ and ∇ are defined by

$$\Delta f(x) := f(x+1) - f(x), \qquad \nabla f(x) := f(x) - f(x-1),$$

and the *q-derivative operator* $\mathbf{D}_q$ is given by

$$\mathbf{D}_q f(x) := \frac{f(qx) - f(x)}{(q-1)x} \qquad (x \neq 0),$$

with $\mathbf{D}_q f(0) := f'(0)$ (provided $f'(0)$ exists).

Any system $\{P_k\}$ of the classical orthogonal polynomials of a discrete variable, or q-classical polynomials, has the orthogonality property,

$$\underset{A}{\overset{B}{\mathfrak{S}}} \{\varrho P_k P_l\}\, dx = h_k \delta_{kl}, \qquad h_k \neq 0, \tag{3.1}$$

where ϱ is the weight function, which satisfies the *Pearson equation*

$$\mathbf{D}^{+}(\sigma\varrho) = \tau\varrho,$$

where σ and τ are polynomials, $\deg\sigma \leq 2$, $\deg\tau = 1$. Here the symbol $\underset{A}{\overset{B}{\mathfrak{S}}}\, dx$ stands for summation $\sum_{x=A}^{B}$ (case DCOP), or q-integral $\int_A^B d_q x$ (case qCOP).

Also, we have the following three-term recurrence relation, structure relations, difference equation and difference-recurrence identities [15], [29], [34]:

$$xP_k(x) = \mathfrak{X} P_k(x), \text{ where } \mathfrak{X} := \xi_0(k)\,\mathcal{E}^{-1} + \xi_1(k)\mathcal{I} + \xi_2(k)\,\mathcal{E},$$

$$\mathbf{L}_k P_k \equiv \{\sigma \mathbf{D}^{+}\mathbf{D}^{-} + \tau\mathbf{D}^{+} + \lambda_k \mathbf{I}\} P_k = 0, \qquad \lambda_k := \kappa(k)\left[\kappa(1-k)\sigma''/2 - \tau'\right]; \tag{3.2}$$

$$\sigma^{\pm}\mathbf{D}^{\pm} P_k = \lambda_k \mathcal{D}^{\pm} P_k, \text{ where } \mathcal{D} := \delta_0^{\pm}(k)\,\mathcal{E}^{-1} + \delta_1^{\pm}(k)\mathcal{I} + \delta_2^{\pm}(k)\,\mathcal{E}, \tag{3.3}$$

$$\frac{\sigma^{\pm}(x)}{x - \zeta^{\pm}} \mathcal{Q}^{\pm}_{\zeta^{\pm}}\left(\lambda_k^{-1}\mathbf{D}^{\pm} P_k(x)\right) = \mathcal{P}^{\pm}_{\zeta^{\pm}} P_k(x), \text{ where } \zeta^{\pm} \text{ is any root of } \sigma^{\pm},$$
$$\text{and} \quad \mathcal{P}^{\pm}_{\zeta^{\pm}} := \mathcal{I} + \pi^{\pm}(\zeta^{\pm}; k)\,\mathcal{E}, \qquad \mathcal{Q}^{\pm}_{\zeta^{\pm}} := \vartheta^{\pm}(k)\,\mathcal{I} + \omega^{\pm}(\zeta^{\pm}; k)\,\mathcal{E}.$$

Here

$$\vartheta^{\pm}(k) := \xi_0(k)/\delta_0^{\pm}(k),$$

$$\kappa(m) := \begin{cases} m & \text{(DCOP)}, \\ \dfrac{q^m - 1}{q-1} & \text{(qCOP)}, \end{cases} \qquad \sigma^{-} := \sigma, \qquad \sigma^{+} := \begin{cases} \sigma + \tau & \text{(DCOP)}, \\ \sigma + x(q-1)\tau & \text{(qCOP)}. \end{cases} \tag{3.4}$$

The *Fourier coefficients* are now defined by

$$a_k[f] := \frac{1}{h_k} \underset{A}{\overset{B}{\mathfrak{S}}} \{\varrho p_k f\}\, dx. \tag{3.5}$$

3.2. **Basic operators.** Given roots ζ^+ and ζ^- of the polynomials σ^+ and σ^-, respectively, let us define for $i=0,1,\ldots$ the operators $\mathcal{P}^{\pm(i)}_{\zeta^\pm}$ and $\mathcal{Q}^{\pm(i)}_{\zeta^\pm}$ of the form

$$\begin{aligned}\mathcal{P}^{\pm(i)}_{\zeta^\pm} &:= \mathcal{I} + \pi^\pm_i(\zeta^\pm;k)\,\mathcal{E},\\ \mathcal{Q}^{\pm(i)}_{\zeta^\pm} &:= \vartheta^\pm(k)\,\mathcal{I} + \omega^\pm_i(\zeta^\pm;k)\,\mathcal{E},\end{aligned}\tag{3.6}$$

in the following recursive way:

$$\pi^\pm_0(\zeta;k) := \pi^\pm(\zeta;k), \qquad \omega^\pm_0(\zeta;k) := \omega^\pm(\zeta;k),$$

$$\begin{cases}\pi^\pm_i(\zeta;k) := \eta^\pm(k+i)\,\omega^\pm_i(\zeta^\pm;k),\\ \omega^\pm_i(\zeta^\pm;k) := \dfrac{1-\vartheta^\pm(k)\eta^\pm(k+i-1)}{1-\vartheta^\pm(k+1)\eta^\pm(k+i)}\omega^\pm_{i-1}(\zeta^\pm;k)\end{cases} \qquad (i\ge 1),$$

where

$$\eta^\pm(k) := \delta^\pm_2(k+1)/\xi_2(k+1).$$

It can be checked that

$$\begin{aligned}&\mathcal{P}^{\pm(0)}_{\zeta^\pm} = \mathcal{P}^\pm_{\zeta^\pm}, \qquad \mathcal{Q}^{\pm(0)}_{\zeta^\pm} = \mathcal{Q}^\pm_{\zeta^\pm},\\ &\mathcal{P}^{\pm(i)}_{\zeta^\pm}\mathcal{Q}^{\pm(i-1)}_{\zeta^\pm} = \mathcal{Q}^{\pm(i)}_{\zeta^\pm}\mathcal{P}^{\pm(i-1)}_{\zeta^\pm} \qquad (i=1,2,\ldots).\end{aligned}$$

Further, let

$$\begin{cases}\mathcal{S}^{\pm(i,j)}_{\zeta^\pm} := \begin{cases}\mathcal{I} & (i<j),\\ \mathcal{P}^{\pm(i)}_{\zeta^\pm}\mathcal{P}^{\pm(i-1)}_{\zeta^\pm}\ldots\mathcal{P}^{\pm(j)}_{\zeta^\pm} & (i\ge j\ge 0)\end{cases}\\ \mathcal{Z}^{\pm(i)}_{\zeta^\pm} := \mathcal{S}^{\pm(i-1,0)}_{\zeta^\pm} \qquad (i\ge 0),\\ \mathcal{U}^{\pm(0)}_{\zeta^\pm} := \mathcal{I},\\ \mathcal{U}^{\pm(i)}_{\zeta^\pm} := \mathcal{Q}^{\pm(i-1)}_{\zeta^\pm}\ldots\mathcal{Q}^{\pm(1)}_{\zeta^\pm}\mathcal{Q}^{\pm(0)}_{\zeta^\pm} \qquad (i\ge 1).\end{cases}$$

Having defined the sequence of operators $\mathcal{P}^{\pm(i)}_{\zeta^\pm}$ $(i=0,1,\ldots)$ corresponding to a given root $\zeta^\pm$, let us define for $i=0,1,\ldots$ the operators $\mathcal{R}^{\pm(i)}_{\zeta^\pm}$ of the form

$$\mathcal{R}^{\pm(i)}_{\zeta^\pm} := \delta^\pm_0(k)\,\mathcal{E}^{-1} + \rho^\pm_i(\zeta;k)\,\mathcal{I},\tag{3.7}$$

where $\delta^\pm_0$ is that of (3.3), in the following recursive way:

$$\begin{aligned}\rho^\pm_0(\zeta^\pm;k) &:= \frac{\delta^\pm_2(k)}{\pi^\pm(\zeta^\pm;k)},\\ \rho^\pm_i(\zeta^\pm;k) &:= \frac{\pi^\pm_{i-1}(\zeta^\pm;k)}{\pi^\pm_i(\zeta^\pm;k)}\rho^\pm_{i-1}(\zeta^\pm;k+1) \qquad (i\ge 1).\end{aligned}$$

It can be checked that

$$\begin{aligned}&\mathcal{R}^{\pm(0)}_{\zeta^\pm}\mathcal{P}^{\pm(0)}_{\zeta^\pm} = \mathcal{D}^\pm,\\ &\mathcal{R}^{\pm(i)}_{\zeta^\pm}\mathcal{P}^{\pm(i)}_{\zeta^\pm} = \mathcal{P}^{\pm(i-1)}_{\zeta^\pm}\mathcal{R}^{\pm(i-1)}_{\zeta^\pm} \qquad (i=1,2,\ldots).\end{aligned}$$

Further, for i, $j\ge 0$, let us define

$$\begin{aligned}\mathcal{M}^{\pm(i,j)}_{\zeta^\pm} &:= \begin{cases}\mathcal{I} & (i>j),\\ \mathcal{R}^{\pm(i)}_{\zeta^\pm}\mathcal{R}^{\pm(i+1)}_{\zeta^\pm}\ldots\mathcal{R}^{\pm(j)}_{\zeta^\pm} & (0\le i\le j),\end{cases}\\ \mathcal{N}^{\pm(i)}_{\zeta^\pm} &:= \mathcal{M}^{\pm(0,i-1)}_{\zeta^\pm}.\end{aligned}$$

Finally, let us define the operators $\mathcal{F}^{\pm(i)}_{\omega^+,\omega^-}$ $(i=0,1,\dots;\ \omega^\pm\in\{\zeta^\pm,\zeta^\pm_*\})$ of the form

$$\mathcal{F}^{\pm(i)}_{\omega^+,\omega^-} := \varphi^\pm(\omega^+,\omega^-;k)\,\mathcal{E}^{-1} + \varphi^\pm_i(\omega^+,\omega^-;k)\,\mathcal{I},$$

where

$$\varphi^\pm(\omega^+,\omega^-;k) := \varepsilon^\pm\,\nu^\pm_{\omega^+,\omega^-}\,\frac{\delta^\mp_0(k)}{\gamma^\pm(\omega^+,\omega^-;k-1)},$$

and

$$\gamma^\pm(\omega^+,\omega^-;k) := \vartheta^\pm(k) + \nu^\pm_{\omega^+,\omega^-}, \tag{3.8}$$

$$\nu^\pm_{\omega^+,\,\omega^-} := \frac{2}{(\sigma^\pm)''}\,\frac{\tau(\omega^\mp_*)}{\omega^\mp_* - \omega^\pm_*}, \tag{3.9}$$

in the following recursive way:

$$\varphi^\pm_0(\omega^+,\omega^-;k) := \varepsilon^\pm\,\nu^\pm_{\omega^+,\omega^-}\,\frac{\rho^\mp_0(\omega^\mp;k)}{\gamma^\pm(\omega^+,\omega^-;k)},$$

$$\varphi^\pm_i(\omega^+,\omega^-;k) := \frac{\pi^\pm_{i-1}(\omega^\pm;k)}{\pi^\pm_i(\omega^\pm;k)}\,\varphi^\pm_{i-1}(\omega^+,\omega^-;k+1) \qquad (i\ge 1).$$

Here

$$\varepsilon^+ := 1, \quad \varepsilon^- := \begin{cases} 1 & \textsf{(DCOP)},\\ q & \textsf{(qCOP)}.\end{cases}$$

Obviously, the sequence $\mathcal{F}^{\pm(i)}_{\omega^+,\omega^-}$ is well-defined only in the case when $\omega^+_* \ne \omega^-_*$ (cf. (3.9), (3.8)).

It can be verified that

$$\mathcal{F}^{\pm(0)}_{\omega^+,\omega^-} := \varepsilon^\pm\,\nu^\pm_{\omega^+,\omega^-}\,\mathcal{R}^{\mp(0)}_{\omega^\mp}\left(\frac{1}{\gamma^\pm(\omega^+,\omega^-;k)}\,\mathcal{I}\right),$$

$$\mathcal{F}^{\pm(i)}_{\omega^+,\omega^-}\mathcal{P}^{\pm(i)}_{\omega^\pm} = \mathcal{P}^{\pm(i-1)}_{\omega^\pm}\mathcal{F}^{\pm(i-1)}_{\omega^+,\omega^-} \qquad (i=1,2,\dots),$$

where we use notation of (3.6) and (3.7). Further, let

$$\mathcal{W}^{\pm(i,j)}_{\omega^+,\omega^-} := \begin{cases}\mathcal{I} & (i>j),\\ \mathcal{F}^{\pm(i)}_{\omega^+,\omega^-}\mathcal{F}^{\pm(i+1)}_{\omega^+,\omega^-}\cdots\mathcal{F}^{\pm(j)}_{\omega^+,\omega^-}, & (0\le i\le j),\end{cases}$$

$$\mathcal{Y}^{\pm(i)}_{\omega^+,\omega^-} := \mathcal{W}^{\pm(0,i-1)}_{\omega^+,\omega^-} \qquad (i\ge 0).$$

3.3. **Algorithm.** Let $\{P_k\}$ be any sequence of the classical orthogonal polynomials of a discrete variable, or q-classical polynomials. Let us define the difference operators $\mathbf{U}^\pm$ and $\mathbf{Z}^\pm_{\zeta^\pm}$ (acting on x) by

$$\mathbf{U}^\pm := \frac{1}{\varepsilon^\pm}\sigma^\pm(x)\mathbf{D}^\pm + \tau(x)\mathbf{I},$$

$$\mathbf{Z}^\pm_{\zeta^\pm} := \frac{1}{\varepsilon^\pm}(x-\zeta^\pm)\mathbf{D}^\pm,$$

where $\zeta^\pm$ is any root of the polynomial $\sigma^\pm$.

Algorithm 3.1 ([28, 48]). *Let $\{P_k\}$ be any sequence of the classical orthogonal polynomials of a discrete variable, or q-classical polynomials in the Hahn sense. Let assume that the function $f(x)$ satisfies the following equation:*

$$\mathbf{P}f(x) \equiv \mathbf{P}^+f(x) + \mathbf{P}^-f(x) = g(x), \tag{3.10}$$

where

$$\mathbf{P}^{\pm} f(x) := \sum_{i=0}^{n^{\pm}} w^{\pm}_{n^{\pm},i}(x)\left(\mathbf{D}^{\pm}\right)^{i} f(x), \tag{3.11}$$

and $w^{\pm}_{n^{\pm},i}$ *are polynomials.*

Step 1: *With* $n := n^{+}$, *define the differential operators* $\mathbf{P}^{+}_{i}$ $(i = 0, 1, \ldots, n-1)$ *and* $\mathbf{Q}^{+}_{j}$ $(j = 1, 2, \ldots, n)$, *and the polynomials* $z^{+}_{0}, z^{+}_{1}, \ldots, z^{+}_{n}$ *in the following recursive way.*

For $i = n, n-1, \ldots, 1$, *given the operator* $\mathbf{P}^{+}_{i} = \sum_{j=0}^{i} w^{+}_{ij}\left(\mathbf{D}^{+}\right)^{j}$,

- *represent the leading coefficient* w^{+}_{ii} *in the form*

$$w^{+}_{ii}(x) = \left(\prod_{h=1}^{\alpha_i}\left(\mathbf{E}^{+}\right)^{i-h}\sigma^{+}(x)\right)\cdot\left(\prod_{m=\alpha_i+1}^{\beta_i}\left(\mathbf{E}^{+}\right)^{i-m}(x-\zeta^{+}_{i})\right)\cdot u_i(x)$$

  where u_i *is a polynomial,* ζ^{+}_{i} *is a root of* σ^{+}, *and the integer exponents* α_i, β_i *have maximal possible values such that* $0 \le \alpha_i \le i$, *and* $\alpha_i \le \beta_i \le B_i$, *where* $B_i = i$ *if* $\deg\sigma^{+} = 2$, *and* $B_i = \alpha_i$ *if* $\deg\sigma^{+} < 2$ *(if* $\beta_i = \alpha_i$, *the value of* ζ_i *is inessential);*
- *define*

$$r^{+}_{i} := i - \beta_i, \qquad s^{+}_{i} := \beta_i - \alpha_i, \qquad t^{+}_{i} := \alpha_i,$$
$$z^{+}_{i}(x) := \left(\mathbf{E}^{+}\right)^{-i} u_i(x);$$

- *define the operator* $\mathbf{Q}^{+}_{i}$ *by*

$$\mathbf{Q}^{+}_{i} := \left(\mathbf{D}^{+}\right)^{r^{+}_{i}}\left(\mathbf{Z}^{+}_{\zeta^{+}_{i}}\right)^{s^{+}_{i}}\left(\mathbf{U}^{+}\right)^{t^{+}_{i}};$$

- *define the operator* $\mathbf{P}^{+}_{i-1}$ *of degree* $i-1$ *as*

$$\mathbf{P}^{+}_{i-1} f(x) := \mathbf{P}^{+}_{i} f(x) - \mathbf{Q}^{+}_{i}\left(z^{+}_{i} f\right)(x).$$

Denote $\mathbf{Q}^{+}_{0} := \mathbf{I}$, $z^{+}_{0} := w_{0,0}$.

Step 2: *Repeat Step 1 with the superscript '+' changed for '−'.*

Step 3: *With*

$$n := n^{+}, \quad \zeta := \zeta^{+}, \quad \zeta_* := \zeta^{+}_{*}, \quad \zeta_i := \zeta^{+}_{i}, \quad r_i := r^{+}_{i}, \quad s_i := s^{+}_{i}, \quad t_i := t^{+}_{i}, \quad z_i := z^{+}_{i},$$

define

$$\Omega := \{1, 2, \ldots, n\}, \qquad \Omega_\eta := \{i \in \Omega : \zeta_i = \eta\} \qquad (\eta \in \{\zeta, \zeta_*\}),$$
$$e_\eta := \max_{i\in\Omega_\eta} r_i, \qquad d_\eta := \max_{i\in\Omega_\eta}(r_i + s_i) - e_\eta \qquad (\eta \in \{\zeta, \zeta_*\}),$$
$$\begin{cases} \omega^{+} := \zeta,\ \omega^{+}_{*} := \zeta_* & \text{if} \quad d_\zeta + e_\zeta \ge d_{\zeta_*} + e_{\zeta_*},\\ \omega^{+} := \zeta_*,\ \omega^{+}_{*} := \zeta & \text{if} \quad d_\zeta + e_\zeta < d_{\zeta_*} + e_{\zeta_*},\end{cases}$$
$$e^{+} := \max(e_{\omega^{+}}, d_{\omega^{+}_{*}} + e_{\omega^{+}_{*}}), \qquad d^{+} := d_{\omega^{+}} + e_{\omega^{+}} - e^{+}.$$

Further, define

$$\mathcal{T}^{+} := \mathcal{Z}^{+(d^{+})}_{\omega^{+}}\left(\mathcal{D}^{+}\right)^{e^{+}},$$
$$\mathcal{A}_i := \left(\varepsilon^{+}\right)^{r_i}\mathcal{U}^{+(s_i)}_{\omega^{+}}\left(-\frac{\lambda_k}{\varepsilon^{-}}\mathcal{D}^{-}\right)^{t_i} \qquad (i = 0, 1, \ldots, n),$$
$$\mathcal{B}_i := \mathcal{M}^{+(d_\eta, e_\eta + d_\eta - r_i - 1)}_{\eta}\mathcal{S}^{+(e_\eta + d_\eta - r_i - 1, s_i)}_{\eta} \qquad (i \in \Omega_\eta;\ \eta = \zeta, \zeta_*),$$

$$\mathcal{J}_{\omega^+} := \mathcal{M}^{+(d,d_\omega^+-1)}_{\omega^+}, \qquad \mathcal{J}_{\omega_*^+} := \mathcal{Z}^{+(d)}_{\omega^+}\mathcal{N}^{+(e-e_{\omega_*^+})}_{\omega_*^+}\mathcal{S}^{+(e-e_{\omega_*^+}-1,d_{\omega_*^+})}_{\omega_*^+},$$

$$\mathcal{C}_0 := \mathcal{T}^+, \qquad \mathcal{C}_i := \begin{cases}\mathcal{J}_{\omega^+}\mathcal{B}_i & (i \in \Omega_\omega^+),\\ \mathcal{J}_{\omega_*^+}\mathcal{B}_i & (i \in \Omega_{\omega_*^+}),\end{cases}$$

$$\mathcal{L}^+ := \sum_{i=0}^{n} \mathcal{C}_i \mathcal{A}_i z_i(\mathcal{X}).$$

Step 4: *Repeat Step 3 with the superscript '+' changed for '−', and vice versa.*

Step 5: *If the conditions*

$$\deg\sigma = \deg\sigma^+ = 2, \ \textit{and} \ (\zeta^+ - \zeta_*^-)(\zeta^- - \zeta_*^+) \neq 0$$

are satisfied, go to Step 6. Otherwise,

Step 5a: *if $d^+ \geq d^-$, replace the operators $\mathcal{T}^-$ and $\mathcal{L}^-$ by*

$$\mathcal{T}^- := \left(\mathcal{D}^-\right)^{d^-+e^-}, \qquad \mathcal{L}^- := \mathcal{N}^{-(d^-)}_{\zeta^-}\mathcal{L}^-,$$

and set

$$e^- := d^- + e^-, \qquad d^- := 0,$$

while

Step 5b: *if $d^+ < d^-$, replace the operators $\mathcal{T}^+$ and $\mathcal{L}^+$ by*

$$\mathcal{T}^+ := \left(\mathcal{D}^+\right)^{d^++e^+}, \qquad \mathcal{L}^+ := \mathcal{N}^{+(d^+)}_{\zeta^+}\mathcal{L}^+,$$

and set

$$e^+ := d^+ + e^+, \qquad d^+ := 0.$$

Step 6: *Define*

$$(**) \qquad \mathcal{T} := \mathcal{Z}\mathcal{Y}^{+(\delta)}_{\omega^+,\omega_*^-}\mathcal{Z}^{+(\delta)}_{\omega^+}\left(\mathcal{D}^-\right)^{e^-}\left(\mathcal{D}^+\right)^{e^+} \qquad (\delta := \min(d^+, d^-)),$$

$$\mathcal{L} = \mathcal{C}^+\mathcal{L}^+ + \mathcal{C}^-\mathcal{L}^-,$$

where

$$\mathcal{Z} := \begin{cases}\mathcal{Z}^{+(d^+-d^-)}_{\omega^+} & (d^+ \geq d^-),\\ \mathcal{Z}^{-(d^--d^+)}_{\omega^-} & (d^+ < d^-),\end{cases}$$

$$\mathcal{C}^+ := \begin{cases}\mathcal{W}^{+(d^+-d^-,d^+-1)}_{\omega^+,\omega_*^-}\left(\Delta^{+(d^+)}_{\omega^+}\right)^{e^-} & (d^+ \geq d^-),\\ \mathcal{Z}\mathcal{Y}^{+(d^+)}_{\omega^+,\omega_*^-}\left(\Delta^{+(d^+)}_{\omega^+}\right)^{e^-} & (d^+ < d^-),\end{cases}$$

$$\mathcal{C}^- := \begin{cases}(\varepsilon^-)^{e^+e^-}\mathcal{Z}\mathcal{Y}^{-(d^-)}_{\omega_*^+,\omega^-}\left(\Delta^{-(d^-)}_{\omega^-}\right)^{e^+} & (d^+ \geq d^-),\\ (\varepsilon^-)^{e^+e^-}\mathcal{W}^{-(d^--d^+,d^--1)}_{\omega_*^+,\omega^-}\left(\Delta^{-(d^-)}_{\omega^-}\right)^{e^+} & (d^+ < d^-).\end{cases}$$

Here

$$\Delta^{\pm(i)}_{\omega^\pm} := \mathcal{R}^{\pm(i)}_{\omega^\pm}\mathcal{P}^{\pm(i)}_{\omega^\pm} \pm \mathcal{I} \qquad (i \geq 0),$$

in the case DCOP, *and*

$$\Delta^{\pm(i)}_{\omega^\pm} := \begin{cases}\mathcal{D}^\mp & (i = 0),\\ \left[\mathcal{P}^{\pm(i-1)}_{\omega^\pm} \pm (q-1)\mathcal{Q}^{\pm(i-1)}_{\omega^\pm}\right]\mathcal{R}^{\pm(i-1)}_{\omega^\pm} \pm \omega^\pm(q-1)\,\mathcal{I} & (i \geq 1),\end{cases}$$

in the case qCOP.

Then the coefficients $a_k[f]$ satisfy the recurrence relation

$$\mathcal{L}(h_k a_k[f]) = \varphi(k),$$

where

$$\mathcal{L} := \mathcal{C}^+\mathcal{L}^+ + \mathcal{C}^-\mathcal{L}^-$$
$$\varphi(k) := \mathcal{T}\left(h_k a_k[g]\right).$$

4. Examples

4.1. **Very classical orthogonal polynomials.** Let the fuction $f(x) = (1+x)e^{-x-1}$ be expanded in the Jacobi polynomials $P_k^{(\alpha,\beta)}$,

$$f = \sum_{k=0}^{\infty} a_k[f] P_k^{(\alpha,\beta)}.$$

This function satisfies the first-order differential equation

$$(x+1)f'(x) + xf(x) = 0.$$

A Maple session initialized by

```
>read(`jacobi.mpl`):
>Jacobi([x,x+1]);
```

brings the following results:

$$\begin{aligned}
A[-1] &= -(2\,k+5+\alpha+\beta)\,(2\,k+\alpha+\beta+2)\,(2\,k+\alpha+\beta+1)\,(2\,k+\alpha+\beta)\\
&\quad (2\,k+4+\alpha+\beta)^2\,(2\,k+\alpha+\beta+3)^2\\
A[0] &= -(k\alpha^2-\alpha^2+2k\alpha\beta+3k\alpha+4k^2\alpha-\alpha+4k^2\beta+4k^2+k\beta^2-2k+\beta^2+4k^3-\beta+k\beta)\\
&\quad (2k+\alpha+\beta+3)(2k+5+\alpha+\beta)(2k+\alpha+\beta+2)(2k+\alpha+\beta+1)(2k+4+\alpha+\beta)^2\\
A[1] &= -2(\alpha^3+10\alpha^2+3\beta\alpha^2+5k\alpha^2+10k\alpha\beta+16\alpha\beta+27k\alpha+3\alpha\beta^2+21\alpha\\
&\quad +8k^2\alpha+6\beta^2+\beta^3+5k\beta^2+25k\beta+12+20k^2+4k^3+8k^2\beta+17\beta+30k)\\
&\quad (2k+4+\alpha+\beta)(k+1)(2k+\alpha+\beta+3)(k+\beta+1)(2k+5+\alpha+\beta)(2k+\alpha+\beta)\\
A[2] &= 8(k+2+\alpha)(k+\alpha+\beta+2)(k+1)(k+2)(k+\beta+1)(k+2+\beta)(2k+\alpha+\beta+1)\\
&\quad (2k+\alpha+\beta)\\
w &= (2\,k+5+\alpha+\beta)\,(2\,k+\alpha+\beta+2)\,(2\,k+\alpha+\beta+1)\,(2\,k+\alpha+\beta)\\
&\quad (2\,k+4+\alpha+\beta)^2\,(2\,k+\alpha+\beta+3)^2
\end{aligned}$$

$$s=0,\ \omega=-1, d=1,\ e=0$$
$$T = \frac{E^s\,{Z_\omega}^{(d)}\,D^e}{h(k)}$$

$$[-1,\,2,\,A,(2\,k+5+\alpha+\beta)\,(2\,k+\alpha+\beta+2)\,(2\,k+\alpha+\beta+1)\,(2\,k+\alpha+\beta)$$
$$(2\,k+4+\alpha+\beta)^2\,(2\,k+\alpha+\beta+3)^2]$$

Thus, we have obtained the third-order recurrence relation

$$\sum_{i=-1}^{2} A[i]a_{k+i}[f] = 0$$

with the above given coefficients $A[i]$.

4.2. **Classical orthogonal polynomials of a discrete variable.** Let $f(x) = (x-n+1)_n$ (Pochhammer's polynomial) be represented in terms of the Hahn polynomials,

$$f(x) = \sum_{k=0}^{n} a_k[f] Q_k(x;\alpha,\beta,N) \qquad (n < N).$$

This function satisfies the first-order difference equation

$$x\mathbf{D}^- f(x) - nf(x) = 0$$

with $\mathbf{D}^- f(x) \equiv \nabla f(x) = f(x) - f(x-1)$.

A Maple session initialized by

```
>read(`hahn.mpl`):
>Hahn([-n,x],[]);
```

yields

$$\begin{aligned} A[0] &= (-n+k)\,(2\,k+\alpha+\beta+2)\,(2\,k+\alpha+\beta+3) \\ A[1] &= -(n+2+k+\alpha+\beta)\,(k+1)\,(k-N+1)\,(k+\beta+1) \\ w &= (2\,k+\alpha+\beta+2)\,(2\,k+\alpha+\beta+3) \end{aligned}$$

$$s=0,\ \varepsilon=0,\ \zeta=N-1,\ \eta=N+\alpha,\ d=1,\ t=0,\ u=0,\ v=0$$

$$T = \frac{E^s\, {Z_\varepsilon}^{-(d)}\, {Y_{\zeta,\eta}}^{+(t)}\, {Z_\zeta}^{+(t)}\, (D^-)^u\, (D^+)^v}{h(k)}$$

$$[0,\ 1,\ A,\ (2\,k+\alpha+\beta+2)\,(2\,k+\alpha+\beta+3)]$$

Thus, we have obtained the first-order recurrence relation

$$A[0]a_k[f] + A[1]a_{k+1}[f] = 0$$

with the above given coefficients $A[0]$, $A[1]$.

4.3. **q-Classical orthogonal polynomials.** Let the polynomial $f(x) = x^n$ be represented in terms of the little q-Jacobi polynomials,

$$f(x) = \sum_{k=0}^{n} a_k[f] p_k(x;a,b|q).$$

This function satisfies the first-order q-difference equation

$$x\mathbf{D}^+\,(f)(x) - \frac{q^n-1}{q-1} f(x) = 0$$

with $\mathbf{D}^+ \equiv \mathbf{D}_q$. A Maple session started by the commands

```
>read(`l_jacobi.mpl`):
>l_qJacobi([],[-qnum(n),x]);
```

yields

$$\begin{aligned} A[0] &= -\frac{q^n - q^k}{q-1} \\ A[1] &= \frac{(a\,q^k\,q-1)\,(q^k\,q-1)\,q^k\,(a\,b\,q^k\,q^2\,q^n-1)}{(a\,b\,(q^k)^2\,q^3-1)\,(q-1)\,(a\,b\,(q^k)^2\,q^2-1)} \end{aligned}$$

$$s=0,\ \zeta=0,\ \eta=0,\ d=1,\ t=0,\ u=0,\ v=0$$

$$T = \frac{E^s\, {Z_\zeta}^{+(d)}\, {Y_{\zeta,\eta}}^{+(t)}\, {Z_\zeta}^{+(t)}\, (D^-)^u\, (D^+)^v}{h(k)}$$

$$[0,\ 1,\ A]$$

Thus, we have obtained the first-order recurrence relation

$$A[0]a_k[f] + A[1]a_{k+1}[f] = 0$$

with the above given coefficients $A[0]$, $A[1]$.

Appendix A. Data of the classical orthogonal polynomials

Table 1. Hypergeometric series representations of the monic very classical orthogonal polynomials

Family	Hypergeometric series
Jacobi	$P_k^{(\alpha,\beta)}(x) = (-1)^k \binom{k+\beta}{k} {}_2F_1\left(\begin{matrix} -k,\ k+\alpha+\beta+1 \\ \beta+1 \end{matrix} \middle\vert \dfrac{1+x}{2}\right)$
Laguerre	$L_k^{\alpha}(x) = (\alpha+1)_k(-1)^k\, {}_1F_1\left(\begin{matrix} -k \\ \alpha+1 \end{matrix} \middle\vert x\right)$
Hermite	$H_k(x) = x^k\, {}_2F_0\left(\begin{matrix} -k/2,\ -k/2+1/2 \\ — \end{matrix} \middle\vert -\dfrac{1}{x^2}\right)$
Bessel	$Y_k^{\alpha}(x) = \dfrac{2^k}{(k+\alpha+1)_k}\, {}_2F_0\left(\begin{matrix} -k,\ k+\alpha+1 \\ — \end{matrix} \middle\vert -\dfrac{x}{2}\right)$

Table 2. Hypergeometric series representations of the monic classical orthogonal polynomials of a discrete variable

Family	Hypergeometric series
Hahn	$Q_k(x;\alpha,\beta,N) = \dfrac{(\beta+1)_k(1-N)_k}{(k+\alpha+\beta+1)_k}\, {}_3F_2\left(\begin{matrix} -k,\ -x,\ k+\alpha+\beta+1 \\ \beta+1,\ 1-N \end{matrix} \middle\vert 1\right)$
Meixner	$M_k^{(\beta,c)}(x) = (\beta)_k \left(\dfrac{c}{c-1}\right)^k {}_2F_1\left(\begin{matrix} -k,\ -x \\ \beta \end{matrix} \middle\vert 1-\dfrac{1}{c}\right)$
Krawtchouk	$K_k^{(p)}(x,N) = (-N)_k p^k \binom{N}{k} {}_2F_1\left(\begin{matrix} -k,\ -x \\ -N \end{matrix} \middle\vert -\dfrac{1}{p}\right)$
Charlier	$C_k^{(a)}(x) = (-a)^k\, {}_2F_0\left(\begin{matrix} -k,\ -x \\ — \end{matrix} \middle\vert -\dfrac{1}{a}\right)$

Table 3. Basic hypergeometric series representations of the monic q-classical orthogonal polynomials [11, 15, 32]

Family	Basic hypergeometric series
big q-Jacobi	$P_k(x;a,b,c;q)=\dfrac{(aq;q)_k(cq;q)_k}{(abq^{k+1};q)_k}\,{}_3\phi_2\left(\begin{matrix} q^{-k},\, abq^{k+1},\, x \\ aq,\, cq \end{matrix}\,\middle\vert\, q;q\right)$
big q-Laguerre	$P_k(x;a,b;q)=(aq;q)_k(bq;q)_k\,{}_3\phi_2\left(\begin{matrix} q^{-k},\, 0,\, x \\ aq,\, bq \end{matrix}\,\middle\vert\, q;q\right)$
q-Krawtchouk	$K_k(x;p,N;q)=\dfrac{(q^{-N};q)_k}{(-pq^{-k};q)_k}\,{}_3\phi_2\left(\begin{matrix} q^{-k},\, -pq^{k},\, x \\ q^{-N},\, 0 \end{matrix}\,\middle\vert\, q;q\right)$
little q-Jacobi	$p_k(x;a,b\vert q)=(-1)^k q^{\binom{k}{2}}\dfrac{(aq;q)_k}{(abq^{k+1};q)_k}\,{}_2\phi_1\left(\begin{matrix} q^{-k},\, abq^{k+1} \\ aq \end{matrix}\,\middle\vert\, q;qx\right)$
q-Meixner	$M_k(x;b,c;q)=(-c)^k q^{-k^2}(bq;q)_k\,{}_2\phi_1\left(\begin{matrix} q^{-k},\, x \\ bq \end{matrix}\,\middle\vert\, q;-\dfrac{q^{k+1}}{c}\right)$
alternative q-Charlier	$K_k(x;a;q)=(-1)^k q^{\binom{k}{2}}(aq;q)_k^{-1}\,{}_2\phi_1\left(\begin{matrix} q^{-k},\, -aq^{k} \\ 0 \end{matrix}\,\middle\vert\, q;qx\right)$
little q-Laguerre/Wall	$L_k^{(a)}(x;q)=(-1)^k q^{\binom{k}{2}}(aq;q)_k\,{}_2\phi_1\left(\begin{matrix} q^{-k},\, 0 \\ aq \end{matrix}\,\middle\vert\, q;qx\right)$
Al-Salam&Carlitz I	$U_k^{(a)}(x;q)=(-a)^k q^{\binom{k}{2}}\,{}_2\phi_1\left(\begin{matrix} q^{-k},\, 1/x \\ 0 \end{matrix}\,\middle\vert\, q;qx/a\right)$
discrete q-Hermite	$h_k(x;q)=q^{\binom{k}{2}}\,{}_2\phi_1\left(\begin{matrix} q^{-k},\, 1/x \\ 0 \end{matrix}\,\middle\vert\, q;-qx\right)$
Al-Salam&Carlitz II	$V_k^{(a)}(x;q)=(-a)^k q^{-\binom{k}{2}}\,{}_2\phi_0\left(\begin{matrix} q^{-k},\, x \\ - \end{matrix}\,\middle\vert\, q;\dfrac{q^k}{a}\right)$
Stieltjes-Wigert	$S_k(x;q)=(-1)^k q^{-k^2}\,{}_1\phi_1\left(\begin{matrix} q^{-k} \\ 0 \end{matrix}\,\middle\vert\, q;-q^{k+1}x\right)$

TABLE 4. Data for the monic Jacobi and Bessel polynomials

	JACOBI	BESSEL
$\sigma(x)$	x^2-1	x^2
$\tau(x)$	$(\gamma+1)x+\delta$	$(\alpha+2)x+2$
λ_k	$-k(k+\gamma)$	$-k(k+\alpha+1)$
h_k	$2^{2k+\gamma}\dfrac{k!\Gamma(k+\alpha+1)\Gamma(k+\beta+1)}{\Gamma(2k+\gamma+1)(k+\gamma)_k}$	$\dfrac{(-1)^{k+1}2^{2k}k!}{(\alpha+1)_{2k+1}(k+\alpha+1)_k}$
$\xi_0(k)$	$\dfrac{4k(k+\alpha)(k+\beta)(k+\gamma-1)}{(2k+\gamma-2)_3(2k+\gamma-1)}$	$\dfrac{-4k(k+\alpha)}{(2k+\alpha)(2k+\alpha-1)_3}$
$\xi_1(k)$	$-\dfrac{(\gamma-1)\delta}{(2k+\gamma-1)(2k+\gamma+1)}$	$-\dfrac{2\alpha}{(2k+\alpha)(2k+\alpha+2)}$
$\xi_2(k)$	1	1
$\delta_0(k)$	$\dfrac{4(k+\alpha)(k+\beta)(k+\gamma-1)}{(2k+\gamma-2)_3(2k+\gamma-1)}$	$-\dfrac{4(k+\alpha)}{(2k+\alpha)(2k+\alpha-1)_3}$
$\delta_1(k)$	$\dfrac{2\delta}{(2k+\gamma-1)(2k+\gamma+1)}$	$\dfrac{4}{(2k+\alpha)(2k+\alpha+2)}$
$\delta_2(k)$	$-\dfrac{1}{k+\gamma}$	$-\dfrac{1}{k+\alpha+1}$

Note: $\gamma := \alpha+\beta+1$, $\delta := \alpha-\beta$.

TABLE 5. Data for the monic Laguerre and Hermite polynomials

	LAGUERRE	HERMITE
$\sigma(x)$	x	1
$\tau(x)$	$1+\alpha-x$	$-2x$
λ_k	k	$2k$
h_k	$k!\Gamma(k+\alpha+1)$	$\sqrt{\pi}2^{-k}k!$
$\xi_0(k)$	$k(k+\alpha)$	$\dfrac{k}{2}$
$\xi_1(k)$	$2k+\alpha+1$	0
$\xi_2(k)$	1	1
$\delta_0(k)$	$k+\alpha$	$\dfrac{1}{2}$
$\delta_1(k)$	1	0
$\delta_2(k)$	0	0

TABLE 6. Data for the monic Hahn polynomials

$\sigma(x)$	$x(N+\alpha-x)$
$\tau(x)$	$(\beta+1)(N-1)-(\gamma+1)x$
$\sigma^+(x)$	$(x+\beta+1)(N-1-x)$
λ_k	$k(k+\gamma)$
h_k	$\dfrac{k!\Gamma(k+\alpha+1)\Gamma(k+\beta+1)(2k+\gamma+1)_{N-k-1}}{(k+\gamma)_k(N-k-1)!}$
$\xi_0(k)$	$\dfrac{k(N-k)(k+\alpha)(k+\beta)(k+\gamma-1)(k+\gamma+N-1)}{(2k+\gamma-2)_2(2k+\gamma-1)_2}$
$\xi_1(k)$	$\dfrac{\alpha-\beta+2N-2}{4}+\dfrac{(\beta^2-\alpha^2)(\gamma+2N-1)}{4(2k+\gamma-1)(2k+\gamma+1)}$
$\xi_2(k)$	1
$\delta_0^+(k)=\delta_0^-(k)$	$\dfrac{(k+\alpha)(k+\beta)(k+\gamma-1)(N-k)(k+\gamma+N-1)}{(2k+\gamma-2)_2(2k+\gamma-1)_2}$
$\delta_1^+(k)=\delta_1^-(k)-1$	$-\dfrac{[2k(k+\gamma)+(\gamma-\alpha)(\gamma-1)-N(\alpha-\beta)]}{(2k+\gamma-1)(2k+\gamma+1)}$
$\delta_2^+(k)=\delta_2^-(k)$	$-\dfrac{1}{k+\gamma}$
$\omega^\pm(\zeta^\pm;k)$	$\dfrac{\delta_1^\pm(k)\xi_0(k)/\delta_0^\pm(k)+\zeta^\pm-\xi_1(k)}{\xi_0(k+1)\delta_2^\pm(k+1)/\xi_2(k+1)-\delta_0^\pm(k+1)}$
$\pi^\pm(\zeta^\pm;k)$	$\delta_2^\pm(k+1)\omega^\pm(\zeta^\pm;k)/\xi_2(k+1)$
$\rho^\pm(\zeta^\pm;k)$	$\delta_2^\pm(k)/\pi^\pm(\zeta^\pm;k)$

Notes: $\gamma:=\alpha+\beta+1$; ζ^+ (resp., ζ^-) is any root of the polynomial σ^+ (resp., σ).

Table 7. Data for the monic Charlier, Meixner and Krawtchouk polynomials

	Charlier	Meixner	Krawtchouk
$\sigma(x)$	x	x	x
$\tau(x)$	$a-x$	$\beta c+(c-1)x$	$(1-p)^{-1}(Np-x)$
$\sigma^{+}(x)$	a	$c(x+\beta)$	$p(1-p)^{-1}(N-x)$
λ_k	k	$(1-c)k$	$(1-p)^{-1}k$
h_k	$k!a^k$	$\dfrac{k!(\beta)_k c^k}{(1-c)^{\beta+2k}}$	$\dfrac{N!k!}{(N-k)!}p^k(1-p)^k$
$\xi_0(k)$	ak	$\dfrac{ck(k+\beta-1)}{(1-c)^2}$	$p(1-p)k(N-k+1)$
$\xi_1(k)$	$k+a$	$\dfrac{(c+1)k+\beta c}{1-c}$	$k+p(N-2k)$
$\xi_2(k)$	1	1	1
$\delta_0^{+}(k)=\delta_0^{-}(k)$	a	$\dfrac{c(k+\beta-1)}{(1-c)^2}$	$p(1-p)(N+1-k)$
$\delta_1^{+}(k)=\delta_1^{-}(k)-1$	0	$\dfrac{c}{1-c}$	$-p$
$\delta_2^{+}(k)=\delta_2^{-}(k)$	0	0	0

TABLE 8. Data of the q-classical orthogonal polynomials: coefficients σ and τ of the equation (3.2), and σ^+ (cf. (3.4)) [31, 32]

$p_k(x;q)$	$\sigma(x)$	$\tau(x)$	$\sigma^+(x)$
$P_k(x;a,b,c;q)$ (big q-Jacobi)	$(x-aq)(x-cq)$	$\frac{abq^2(x-1)-x+cq-aq(cq-1)}{q-1}$	$aq^2(x-1)(bx-c)$
$P_k(x;a,b;q)$ (big q-Laguerre)	$(x-aq)(x-bq)$	$\frac{bq-x-aq(bq-1)}{q-1}$	$abq^2(1-x)$
$K_k(x;p,N;q)$ (q-Krawtchouk)	$x(x-q^{-N})$	$\frac{pq(1-x)-x+q^{-N}}{q-1}$	$pqx(1-x)$
$p_k(x;a,b\vert q)$ (little q-Jacobi)	$x(x-1)$	$\frac{aq(bqx-1)+1-x}{q-1}$	$aqx(bqx-1)$
$M_k(x;b,c;q)$ (q-Meixner)	$c(x-bq)$	$\frac{q(x-1)+c(bq-1)}{q-1}$	$q(x-1)(x+bc)$
$K_k(x;a;q)$ (alternative q-Charlier)	$x(x-1)$	$\frac{1-(aq+1)x}{q-1}$	$-aqx^2$
$L_k^{(a)}(x;q)$ (little q-Laguerre/Wall)	$x(x-1)$	$\frac{x-1+aq}{q-1}$	$-aqx$
$U_k^{(a)}(x;q)$ (Al-Salam&Carlitz I)	$(x-1)(x-a)$	$\frac{x-a-1}{1-q}$	a
$h_k(x;q)$ (discrete q-Hermite)	x^2-1	$\frac{x}{1-q}$	-1
$V_k^{(a)}(x;q)$ (Al-Salam&Carlitz II)	a	$\frac{x-a-1}{q-1}$	$(x-1)(x-a)$
$S_k(x;q)$ (Stieltjes-Wigert)	x	$\frac{qx-1}{q-1}$	qx^2

Table 9. Data of the monic q-classical orthogonal polynomials: coefficients of the difference operators $\mathcal{D}^{\pm}$, $\mathcal{X}$, $\mathcal{P}^{\pm}_{\zeta^{\pm}}$ and $\mathcal{Q}^{\pm}_{\zeta^{\pm}}$ given in terms of σ and τ [29, 31]

$\xi_0(k)$	$\dfrac{q^{k-1}[k]_q R(k-2)\{s_0R(2k-2)-q^{k-1}(s_1[k-1]_q+t_0)\left(q^{k-1}s_2t_0-s_1R(k-1)\right)\}}{R(2k-1)R^2(2k-2)R(2k-3)}$
$\xi_1(k)$	$q^k\dfrac{s_2q^{k-1}[k]_q-R(k-2)-q^{-1}s_1[2]_q[k]_qR(k-1)}{R(2k)R(2k-2)}$
$\xi_2(k)$	1
$\delta_0^+(k)$	$-q^{1-k}R(k-1)\xi_0(k)/\lambda_k$
$\delta_1^+(k)$	$-\dfrac{qt_0}{(q+1)\lambda_k}-\dfrac{q\xi_1(k)\{(q-1)[-k]_qR(k-1)+t_1\}}{(q+1)\lambda_k}$
$\delta_2^+(k)$	$s_2[k]_q/\lambda_k$
$\delta_i^-(k)$	$\delta_i^+(k)+(q-1)\xi_i(k)\quad(i=0,1,2)$
$\vartheta^+(k)$	$-q^{k-1}\lambda_k/R(k-1)$
$\vartheta^-(k)$	$1/[q-1-q^{1-k}R(k-1)/\lambda_k]$
$\omega^{\pm}(\zeta^{\pm};k)$	$\dfrac{\delta_1^{\pm}(k)\xi_0(k)/\delta_0^{\pm}(k)+\zeta^{\pm}-\xi_1(k)}{\xi_0(k+1)\delta_2^{\pm}(k+1)/\xi_2(k+1)-\delta_0^{\pm}(k+1)}$
$\pi^{\pm}(\zeta^{\pm};k)$	$\delta_2^{\pm}(k+1)\omega^{\pm}(\zeta^{\pm};k)/\xi_2(k+1)$
$\rho^{\pm}(\zeta^{\pm};k)$	$\delta_2^{\pm}(k)/\pi^{\pm}(\zeta^{\pm};k)$

Notes: 1. $t_1:=\tau'$, $t_0:=\tau(0)$, $s_2:=\sigma''/2+(q-1)\tau'$, $s_1:=\sigma'(0)+(q-1)\tau(0)$, $s_0:=\sigma(0)$, $R(k):=s_2[k]_q+t_1$.

2. ζ^+ (resp., ζ^-) is any root of the polynomial $\sigma^+(x):=\sigma(x)+(q-1)x\tau(x)$ (resp., $\sigma^-\equiv\sigma$).

Appendix B. Maple programs

B.1. Expansions in the Jacobi polynomials.

```
# Constructing recurrence equation for Jacobi series coefficients
# of a solution of a linear differential equation.
#
####################################################################
#
# Version 1.0 for MapleV release 4
#
# Pawel Wozny (Pawel.Wozny@ii.uni.wroc.pl), 07/02/2001
#
####################################################################
#
# Let f(x) be a solution of a linear differential equation
#
#                           n
#                         -----
#                          \             i
# (1)                       )    v_i(x) D  f(x) = g(x),
#                          /
#                         -----
#                         i = 0
#                                       d
# where v_i(x) are polynomials, and D:= --.
#                                       dx
# Let a_k[f] be the Fourier coefficients of the function f(x) in the
# expansion
#                                 -----
#                                  \
# (2)                      f(x) =   )   a_k[f] P_k(x),
#                                  /
#                                 -----
#                                   k
#
# where P_k(x) is the k-th monic Jacobi polynomial. This program finds
# recurrence relation for a_k[f] of the form
#
#                          r + d
#                          -----
#                           \
# (3)                        )    A_i(k) a_{i+r}[f] = B(k).
#                           /
#                          -----
#                          i = -r
#
# under assumption that a_k[g] are known explicitly. Here r is a natural
# number and d  {0,1}.
#
# Main procedure
# ~~~~~~~~~~~~~~
#   - Jacobi(V)
#
# Input
# ~~~~~
#   - vector V := [v_0(x),v_1(x),...,v_n(x)]
#
# Output
# ~~~~~~
#   - vector [-r,r+d,A,w], where  A[-r..r+d] is a table of A_l(k)
```

```
#     (l=-r..r+d)
#   - operator T (see (*)) is given on the screen  such that
#
#                      B(k)= w*T(a_k[g])
#
####################################################################
#
# Example:
# ~~~~~~~~
#  Let f(x)=x*exp(x). This function satisfies the first-order differential
#  equation
#
#                            xf'(x)-(x+1)f(x)=0.
#
#  We are loking for the recurrence equation for a_k[f]. We should initialize
#  Maple session as follows:
#
#  >read(`jacobi.mpl`):
#  >Jacobi([-x-1,x]);
#
####################################################################

with(LREtools):

bD:=f->D(unapply(f,x))(x):
bV:=f->(x-zeta)*bD(f)+ni(zeta)*f:
bK:=f->bV(bD(f)):
bU:=f->sigma(x)*bD(f)+tau(x)*f:
bL:=f->bU(bD(f)):
zeta:=1:  zeta2:=-1:
_gamma:=alpha+beta+1:  _delta:=alpha-beta:
ni:=z->if z=1 then alpha+1 else beta+1 fi:
mi:=(q,r,z)->pochhammer(k+ni(z),q+r)/(product(delta0(k+v),v=1..q)):
sigma:=x->x^2-1:  tau:=x->(_gamma+1)*x+_delta:
h:=k->2^(2*k+_gamma)*k!*GAMMA(k+alpha+1)*GAMMA(k+beta+1)/
      (GAMMA(2*k+_gamma+1)*pochhammer(k+_gamma,k)):
lambda:=k->-k*(k+_gamma):
ksi0:=k->4*k*(k+alpha)*(k+beta)*(k+_gamma-1)/(pochhammer(2*k+_gamma-2,3)*
         (2*k+_gamma-1)):
ksi1:=k->-_delta*(_gamma-1)/((2*k+_gamma-1)*(2*k+_gamma+1)):
ksi2:=k->1:
delta0:=k->4*(k+alpha)*(k+beta)*pochhammer(k+_gamma-1,2)/
             (pochhammer(2*k+_gamma-2,3)*(k+_gamma)*(2*k+_gamma-1)):
delta1:=k->2*_delta/((2*k+_gamma-1)*(2*k+_gamma+1)):
delta2:=k->-1/(k+_gamma):
pX:=f->ksi0(k)*shift(f,k,-1)+ksi1(k)*f+ksi2(k)*shift(f,k,1):
pD:=f->delta0(k)*shift(f,k,-1)+delta1(k)*f+delta2(k)*shift(f,k,1):
lam_pI:=f->-lambda(k)*f:
lambda_pD:=f->-lambda(k)*pD(f):
pP:=f->f+zeta/2*pochhammer(2*k+_gamma,2)*pochhammer(2*k+_gamma+1,2)/
        ((k+ni(zeta2))*(k+_gamma)*pochhammer(2*k+_gamma+m,2))*shift(f,k,1):
pR:=f->delta0(k)*shift(f,k,-1)-2*zeta*(k+ni(zeta2)+m-1)/
                              pochhammer(2*k+_gamma+m-1,2)*f:
pS:=proc(z,i,j,f)
  global m,zeta,zeta2;
  local l,S;
   S:=(pP@@0)(f);
    if z=1 then zeta:=z; zeta2:=-1
      else zeta:=z; zeta2:=1 fi;
    for l from j to i
```

```
      do
        m:=l;
        S:=pP(S)
      od;
   S
end:

pZ:=proc(z,n,f)
   pS(z,n,1,f)
end:

pM:=proc(z,i,j,f)
  global m,zeta,zeta2;
  local l,R;
   R:=(pR@@0)(f);
    if z=1 then zeta:=z; zeta2:=-1
      else zeta:=z; zeta2:=1 fi;
    for l from j to i by -1
     do
       m:=l;
       R:=pR(R)
     od;
   R
end:

pN:=proc(z,n,f)
   pM(z,1,n,f)
end:

find_bQi:=proc(i,w)
  local wx,a,b,d,p,r,s,m,omega,eps;
    if simplify(w)=0 then
      RETURN([0,0,0,0,0,0,0])
    fi;
   wx:=unapply(simplify(w),x);
   d:=degree(wx(x),x);
   a:=degree(gcd((x-1)^d,wx(x)),x);
   b:=degree(gcd((x+1)^d,wx(x)),x);
   p:=simplify(wx(x)/((x-1)^a*(x+1)^b));
   r:=min(a,b); s:=max(a,b)-r; m:=floor((i+1)/2); omega:=i mod 2;
   if r=a then eps:=-1 else eps:=1 fi;
   if r>=m then
       [0,0,eps,0,m-omega,omega,simplify(p*(x^2-1)^(r-m)*(x-eps)^s)]
     elif (r+s>=m and m>r) then
       [0,m-r-omega,eps,omega,r,0,simplify(p*(x-eps)^(r+s-m))]
     else
       [i-2*r-2*s,s,eps,0,r,0,p]
    fi
end:

canonical_form:=proc(n,p)
  global zeta;
  local i,j,k,w,bQ,q,P;
   P[n]:=p;
    for k from 0 to n
     do
       w[k]:=coeff(P[n],(bD@@k)(f(x)),1)
     od;
   bQ:=array(0..n);
    for i from n to 1 by -1
```

```
     do
       q:=find_bQi(i,w[i]);
       bQ[i]:=q;
       zeta:=q[3];
       P[i-1]:=P[i]-(bD@@q[1]@bK@@q[2]@bV@@q[4]@bL@@q[5]@bU@@q[6])(q[7]*f(x));
        for j from 0 to i-1
         do
           w[j]:=coeff(P[i-1],(bD@@j)(f(x)),1)
         od
     od;
    bQ[0]:=[0,0,0,0,0,0,factor(simplify(w[0]))];
    bQ
end:

poly_to_oper:=proc(w,f)
  local j,i,p;
    for i from 0 to degree(w,x)
     do
       p[i]:=coeff(w,x,i)*(pX@@i)(f)
     od;
   sum(p[j],j=0..degree(w,x))
end:

integral_canonical_form:=proc(n,Q)
  local Omega,z,z2,t,i,j,l,p,p2,q,q2,e,epsilon,epsilon2,
        pom,O,sO,oO,stp,stm,sm,sp;
   z:=1; z2:=-1;
   Omega[z]:=[]; Omega[z2]:=[]; stp:=0; stm:=0;
   p[z]:=0; p[z2]:=0; q[z]:=0; q[z2]:=0;
    for i from 0 to n
     do
        if is(Q[i][3]=z) then
          Omega[z]:=[op(Omega[z]),i];
          p[z]:=max(p[z],Q[i][1]);
          q[z]:=max(q[z],Q[i][1]+2*Q[i][2]+Q[i][4]);
        else
          Omega[z2]:=[op(Omega[z2]),i];
          p[z2]:=max(p[z2],Q[i][1]);
          q[z2]:=max(q[z2],Q[i][1]+2*Q[i][2]+Q[i][4])
        fi
     od;
   q[z]:=q[z]-p[z];
   q[z2]:=q[z2]-p[z2];
   if is(p[z]+q[z]<p[z2]+q[z2]) then epsilon:=z2; epsilon2:=z
     else epsilon:=z; epsilon2:=z2 fi;
   p2:=max(p[epsilon],p[epsilon2]+q[epsilon2]);
   q2:=p[epsilon]+q[epsilon]-p2;
    for e in [epsilon,epsilon2]
     do
      for j in Omega[e]
       do
         pom:=(lam_pI@@Q[j][5]@lambda_pD@@Q[j][6])(poly_to_oper(Q[j][7],b[k]));
         pom:=mi(Q[j][2],Q[j][4],e)*shift(pZ(-e,Q[j][4],pom),k,Q[j][2]);
         pom:=pM(e,q[e]+1,q[e]+p[e]-Q[j][1],
                 pS(e,q[e]+p[e]-Q[j][1],2*Q[j][2]+Q[j][4]+1,pom));
         sp:=degree(Q[j][7],x)+Q[j][6]-Q[j][2]-Q[j][1]+q[e]+p[e];
         sm:=degree(Q[j][7],x)+Q[j][6]+Q[j][4]-Q[j][2]-Q[j][1]+p[e];
         if is(e=epsilon) then
            O[j]:=pM(e,q2+1,q[e],pom);
            sm:=sm+q[e]-q2
```

```
          else
           O[j]:=pZ(epsilon,q2,pN(e,p2-p[e],
                    pS(e,p2-p[e],q[e]+1,pom)));
           sp:=sp+p2+q2-p[e]-q[e];
           sm:=sm+p2-p[e]
         fi;
       stp:=max(stp,sp);
       stm:=max(stm,sm)
      od
    od;
  s0:=0;
  o0:=sum(O[t],t=0..n);
   for l from -stm to stp
     do
       s0:=s0+factor(simplify(coeff(o0,b[k+l],1)))*b[k+l]
     od;
  [s0,stm,stp,[epsilon,q2,p2]]
end:

Jacobi:=proc(lp)
  options `Copyright 2001 by Pawel Wozny`;
  local Pp,np,Kp,m,p,i,j,l,oper,d,d2,A,B,pom,sp,sm,prze,w,o1,o2,o3;
   np:=nops(lp);
   if np=0 then Pp:=0 else
      np:=np-1; Pp:=sum(simplify(lp[l+1])*(bD@@l)(f(x)),l=0..np)
    fi;
   Kp:=canonical_form(np,Pp);
   oper:=integral_canonical_form(np,Kp);
   sm:=0; sp:=0; d:=1; d2:=0;
   for i from -oper[2] to oper[3]
     do
       pom:=coeff(oper[1],b[k+i],1);
       B[i]:=factor(simplify(pom*h(k+i)/h(k)));
       if B[i]<>0 then
          d:=lcm(d,denom(B[i]));
          d2:=gcd(d2,numer(B[i]));
          sp:=max(sp,i);
          sm:=min(sm,i)
        fi
     od;
   prze:=-sm-floor((sp-sm)/2);
   w:=factor(simplify(d/d2));
   o1:=oper[4][1]; o2:=oper[4][2]; o3:=oper[4][3];
   for j from sm+prze to sp+prze
     do
       A[j]:=0;
       if B[j-prze]<>0 then
          A[j]:=factor(simplify(shift(B[j-prze]*w,k,prze)));
          print(`A[`.j.`]`=A[j])
        fi
     od;
   print('w'=shift(w,k,prze));
   print('s'=prze,'omega'=o1,'d'=o2,'e'=o3);
   print(T=1/'h'(k+prze)*E^'s'*Z['omega']^`(d)`*D^'e');
   [sm+prze,sp+prze,A,shift(w,k,prze)]
end:
```

B.2. **Expansions in the Hahn polynomials.**

```
# Constructing recurrence equation for Hahn series coefficients
# of a solution of a linear difference equation.
```

```
#
####################################################################
#
# Version 1.0 for MapleV release 4
#
# Copyright (c) Pawel Wozny (Pawel.Wozny@ii.uni.wroc.pl), 07/02/2001
#
####################################################################
#
# Let f(x) be a solution of a linear difference equation
#
#             n                            m
#           -----                        -----
#            \               i            \               j
# (1)         )   u_i(x) (D-)  f(x)  +     )   v_j(x) (D+)  f(x) = g(x)
#            /                            /
#           -----                        -----
#           i = 0                        j = 0
#
# where u_i(x), v_j(x) are polynomials and
#
#          (D+)(f(x)):=f(x+1)-f(x),   (D-)(f(x)):=f(x)-f(x-1).
#
# Let a_k[f] be the Fourier coefficients of the function f(x) in the
# expansion
#                                -----
#                                 \
# (2)                   f(x) =     )   a_k[f] Q_k(x),
#                                 /
#                                -----
#                                  k
#
# where Q_k(x) is the k-th monic Hahn polynomial. This program finds
# recurrence relation for a_k[f] of the form
#
#                       r + d
#                       -----
#                        \
# (3)                     )    A_i(k) a_{i+r}[f] = B(k)
#                        /
#                       -----
#                       i = -r
#
# under assumption that a_k[g] are known explicitly. Here r is a natural
# number and d  {0,1}.
#
# Main procedure
# ~~~~~~~~~~~~~~
#   - Hahn(U,V)
#
# Input
# ~~~~~
#   - vectors U := [u_0(x),u_1(x),...,u_n(x)] and
#     V := [v_0(x),v_1(x),...,v_m(x)]
#
# Output
# ~~~~~~
#   - vector [-r,r+d,A,w], where  A[-r..r+d] is a table of A_l(k)
#     (l=-r..r+d)
#   - operator T such that  B(k)= w*T(a_k[g])
```

```
#     (see Algorithm 3.1, Step 6, Eq. (**)) is given on the screen
#
####################################################################
#
# Example:
# ~~~~~~~~
#  Let f(x)=product(x+i,i=0..n-1) (Pochhammer's polynomial).
#  This function satisfies the first-order
#  difference equation
#
#                        (x+n-1)(D+)(f(x))-nf(x)=0.
#
#  We are loking for the recurrence equation for a_k[f]. We should initialize
#  Maple session as follows:
#
#  >read(`hahn.mpl`):
#  >Hahn([],[-n,x+n-1]);
#
####################################################################

with(LREtools):
sigma:=x->if R=`+` then -(x+beta+1)*(x-N+1) else x*(N+alpha-x) fi:
tau:=x->(beta+1)*(N-1)-(alpha+beta+2)*x:
bD:=f->if R=`+` then delta(f,x) else f-shift(f,x,-1) fi:
bU:=f->sigma(x)*bD(f,x)+tau(x)*f:
bV:=f->if R=`+` then (x-zeta_p)*bD(f,x)+(N+alpha-zeta_p)*f
        else (x-zeta_m)*bD(f)+(beta+1+zeta_m)*f fi:
zeta_p:=N-1:  zeta_p2:=-beta-1:  zeta_m:=0:  zeta_m2:=N+alpha:
_gamma:=alpha+beta+1:
h:=k->k!*GAMMA(k+alpha+1)*GAMMA(k+beta+1)*pochhammer(2*k+_gamma+1,N-k-1)/
      (pochhammer(k+_gamma,k)*(N-k-1)!):
lambda:=k->k*(k+_gamma):
ksi0:=k->k*(N-k)*(k+alpha)*(k+beta)*(k+_gamma-1)*(k+_gamma+N-1)/
         (pochhammer(2*k+_gamma-2,2)*pochhammer(2*k+_gamma-1,2)):
ksi1:=k->((alpha-beta+2*N-2)/4+(beta^2-alpha^2)*(_gamma+2*N-1)/
                              (4*(2*k+_gamma-1)*(2*k+_gamma+1))):
ksi2:=k->1:
delta_p0:=k->(N-k)*(k+alpha)*(k+beta)*pochhammer(k+_gamma-1,2)*(k+_gamma+N-1)
             /(pochhammer(2*k+_gamma-2,2)*pochhammer(2*k+_gamma-1,2)*(k+_gamma)):
delta_p1:=k->-(2*k*(k+_gamma)+(_gamma-alpha)*(_gamma-1)-
              N*(alpha-beta))/((2*k+_gamma-1)*(2*k+_gamma+1)):
delta_p2:=k->-1/(k+_gamma):
delta_m0:=k->delta_p0(k):
delta_m1:=k->1+delta_p1(k):
delta_m2:=k->delta_p2(k):
pX:=f->ksi0(k)*shift(f,k,-1)+ksi1(k)*f+ksi2(k)*shift(f,k,1):
pD:=f->if R=`+` then
         delta_p0(k)*shift(f,k,-1)+delta_p1(k)*f+delta_p2(k)*shift(f,k,1):
        else
         delta_m0(k)*shift(f,k,-1)+delta_m1(k)*f+delta_m2(k)*shift(f,k,1):
       fi:
lambda_pD:=f->if R=`+` then
                -lambda(k)*(delta_m0(k)*shift(f,k,-1)+
                 delta_m1(k)*f+delta_m2(k)*shift(f,k,1))
               else
                -lambda(k)*(delta_p0(k)*shift(f,k,-1)+
                 delta_p1(k)*f+delta_p2(k)*shift(f,k,1))
              fi:
pP:=f->if R=`+` then
         f+pochhammer(2*k+_gamma,2)*pochhammer(2*k+_gamma+1,2)/
```

```
         (pochhammer(2*k+_gamma+m,2)*(k+_gamma)*(k+_gamma+zeta_p+1)*
         (k-zeta_p2))*shift(f,k,1)
        else
         f-pochhammer(2*k+_gamma,2)*pochhammer(2*k+_gamma+1,2)/
          (pochhammer(2*k+_gamma+m,2)*(k+_gamma)*(k+_gamma+N-zeta_m)*
          (k-N+zeta_m2+1))*shift(f,k,1);
       fi:
pQ:=f->if R=`+` then
         (k+N+alpha-zeta_p)*f-
         pochhammer(2*k+_gamma,2)*pochhammer(2*k+_gamma+1,2)*
         (k+m-zeta_p2-1)/(pochhammer(2*k+_gamma+m,2)*(k+_gamma)*
         (k+_gamma+zeta_p+1)*(k-zeta_p2))*shift(f,k,1)
        else
         (k+beta+zeta_m+1)*f+
         pochhammer(2*k+_gamma,2)*pochhammer(2*k+_gamma+1,2)*(k+m-N+zeta_m2)/
         (pochhammer(2*k+_gamma+m,2)*(k+_gamma)*(k+_gamma+N-zeta_m)*
         (k-N+zeta_m2+1))*shift(f,k,1)
       fi:
pR:=f->if R=`+` then
         delta_p0(k)*shift(f,k,-1)-
         (k+m+_gamma+zeta_p)*(k+m-zeta_p2-1)/pochhammer(2*k+_gamma+m-1,2)*f
        else
         delta_m0(k)*shift(f,k,-1)+
         (k+m+_gamma+N-zeta_m-1)*(k+m-N+zeta_m2)/
         pochhammer(2*k+_gamma+m-1,2)*f
        fi:
pF:=f->if R=`+` then
         delta_p0(k)/(k+zeta_m2-zeta_p-1)*shift(f,k,-1)+
         (k+m+_gamma+zeta_p-zeta_m)/pochhammer(2*k+_gamma+m-1,2)*f;
        else
         delta_m0(k)/(k-zeta_p2+zeta_m-1)*shift(f,k,-1)-
         (k+m+_gamma+zeta_p-zeta_m)/pochhammer(2*k+_gamma+m-1,2)*f
       fi:

find_bQi:=proc(i,w,r)
  local test1,test2,j,wx,q,a,b,z1,z2,s;
   if simplify(w)=0 then
      RETURN([0,0,0,0,0,r])
    fi;
   test1:=true; test2:=true; a:=0; b:=0;
   wx:=unapply(simplify(w),x);
   if r=`+` then
      z1:=-beta-1; z2:=N-1; s:=-1
     else
      z1:=0; z2:=N+alpha; s:=1
    fi;
   for j from 1 to i while (test1 or test2)
     do
       if (test1 and simplify(wx(-s*j+z1+s*i))=0) then
         a:=a+1 else test1:=false fi;
       if (test2 and simplify(wx(-s*j+z2+s*i))=0) then
         b:=b+1 else test2:=false fi
     od;
   q:=simplify(wx(x)/(product(x-s*i-z1+s*k,k=1..a)*
                      product(x-s*i-z2+s*l,l=1..b)));
    if min(a,b)=b then
      [i-a,a-b,z1,b,(-s)^i*(-1)^b*factor(simplify(shift(q,x,s*i))),r]
     else
      [i-b,b-a,z2,a,(-s)^i*(-1)^a*factor(simplify(shift(q,x,s*i))),r]
    fi
```

```
end:

canonical_form:=proc(n,p,r)
  global zeta_p,zeta_m,R;
  local i,j,k,w,bQ,q,s,P;
   R:=r;
   P[n]:=p;
    if R=`+` then
      s:=1 else s:=-1 fi;
    for k from 0 to n
     do
       w[k]:=coeff(P[n],f(x+s*k),1)
     od;
   bQ:=array(0..n);
    for i from n to 1 by -1
     do
       q:=find_bQi(i,w[i],r);
       bQ[i]:=q;
        if r=`+` then zeta_p:=q[3] else zeta_m:=q[3] fi;
       P[i-1]:=P[i]-(bD@@q[1]@bV@@q[2]@bU@@q[4])(q[5]*f(x));
        for j from 0 to i-1
         do
           w[j]:=coeff(P[i-1],f(x+s*j),1)
         od
     od;
    bQ[0]:=[0,0,0,0,factor(simplify(w[0])),r];
    bQ
end:

poly_to_oper:=proc(w,f)
  local i,p,j;
    for i from 0 to degree(w,x)
     do
       p[i]:=coeff(w,x,i)*(pX@@i)(f)
     od;
   sum(p[j],j=0..degree(w,x))
end:

pS:=proc(z,i,j,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,S;
   R:=r;
   S:=(pP@@0)(f);
   if r=`+` then
       if is(z=N-1) then zeta_p:=z; zeta_p2:=-beta-1
        else zeta_p:=z; zeta_p2:=N-1 fi
     else
      if is(z=0) then zeta_m:=z; zeta_m2:=N+alpha
       else zeta_m:=z; zeta_m2:=0 fi
    fi;
    for l from j to i
     do
       m:=l;
       S:=pP(S)
     od;
   S
end:

pZ:=proc(z,n,r,f)
   pS(z,n,1,r,f)
```

```
end:

pU:=proc(z,n,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,Q;
  R:=r;
  Q:=(pQ@@0)(f);
  if r=`+` then
       if is(z=N-1) then zeta_p:=z; zeta_p2:=-beta-1
        else zeta_p:=z; zeta_p2:=N-1 fi
     else
      if is(z=0) then zeta_m:=z; zeta_m2:=N+alpha
       else zeta_m:=z; zeta_m2:=0 fi
    fi;
   for l from 1 to n
     do
       m:=l;
       Q:=pQ(Q)
     od;
   Q
end:

pM:=proc(z,i,j,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,_R;
   R:=r;
   _R:=(pR@@0)(f);
    if r=`+` then
       if is(z=N-1) then zeta_p:=z; zeta_p2:=-beta-1
        else zeta_p:=z; zeta_p2:=N-1 fi
     else
      if is(z=0) then zeta_m:=z; zeta_m2:=N+alpha
       else zeta_m:=z; zeta_m2:=0 fi
    fi;
    for l from j to i by -1
     do
       m:=l;
       _R:=pR(_R)
     od;
   _R
end:

pN:=proc(z,n,r,f)
   pM(z,1,n,r,f)
end:

pW:=proc(zp,zm,i,j,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,V;
   R:=r;
   V:=(pF@@0)(f);
   if is(zp=N-1) then zeta_p:=zp; zeta_p2:=-beta-1
     else zeta_p:=zp; zeta_p2:=N-1 fi;
   if is(zm=0) then zeta_m:=zm; zeta_m2:=N+alpha
     else zeta_m:=zm; zeta_m2:=0 fi;
    for l from j to i by -1
     do
       m:=l;
       V:=pF(V)
     od;
```

```
   V
end:

pY:=proc(zp,zm,n,r,f)
   pW(zp,zm,1,n,r,f)
end:

integrate_canonical_form:=proc(n,Q,r)
  global R;
  local Omega,z,z2,i,j,t,l,p,p2,q,q2,e,epsilon,epsilon2,
        pom,O,sO,oO,stp,stm,sm,sp;
  R:=r;
  if r=`+` then
      z:=N-1; z2:=-beta-1
     else
      z:=0; z2:=N+alpha
    fi;
  Omega[z]:=[]; Omega[z2]:=[]; stp:=0; stm:=0;
  p[z]:=0; p[z2]:=0; q[z]:=0; q[z2]:=0;
  for i from 0 to n
     do
        if is(Q[i][3]=z) then
          Omega[z]:=[op(Omega[z]),i];
          p[z]:=max(p[z],Q[i][1]);
          q[z]:=max(q[z],Q[i][1]+Q[i][2]);
        else
          Omega[z2]:=[op(Omega[z2]),i];
          p[z2]:=max(p[z2],Q[i][1]);
          q[z2]:=max(q[z2],Q[i][1]+Q[i][2])
        fi
     od;
   q[z]:=q[z]-p[z];
   q[z2]:=q[z2]-p[z2];
   if is(p[z]+q[z]<p[z2]+q[z2]) then epsilon:=z2; epsilon2:=z
     else epsilon:=z; epsilon2:=z2 fi;
   p2:=max(p[epsilon],p[epsilon2]+q[epsilon2]);
   q2:=p[epsilon]+q[epsilon]-p2;
   for e in [epsilon,epsilon2]
     do
      for j in Omega[e]
       do
         pom:=(lambda_pD@@Q[j][4])(poly_to_oper(Q[j][5],b[k]));
         pom:=pU(Q[j][3],Q[j][2],r,pom);
         pom:=pM(e,q[e]+1,q[e]+p[e]-Q[j][1],r,
                 pS(e,q[e]+p[e]-Q[j][1],Q[j][2]+1,r,pom));
         sp:=degree(Q[j][5],x)+Q[j][4]+q[e]+p[e]-Q[j][1];
         sm:=degree(Q[j][5],x)+Q[j][4]+p[e]-Q[j][1];
          if is(e=epsilon) then
            O[j]:=pM(e,q2+1,q[e],r,pom);
            sm:=sm+q[e]-q2
           else
            O[j]:=pZ(epsilon,q2,r,pN(e,p2-p[e],r,
                     pS(e,p2-p[e],q[e]+1,r,pom)));
            sp:=sp+p2+q2-p[e]-q[e];
            sm:=sm+p2-p[e]
          fi;
        stp:=max(stp,sp);
        stm:=max(stm,sm)
       od
     od;
```

```
   s0:=0;
   o0:=sum(O[t],t=0..n);
    for l from -stm to stp
      do
        s0:=s0+factor(simplify(coeff(o0,b[k+l],1)))*b[k+l]
      od;
   [epsilon,q2,p2,s0,stm,stp]
end:

p_i_m:=proc(zeta,i,r,pMp,stmp,stpp,eta,j,s,pMm,stmm,stpm)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local p,q,pom,pUp,pUm,z,s0,o0,stm,stp,l,op;
  if is(zeta=N-1) then zeta_p:=zeta; zeta_p2:=-beta-1
      else zeta_p:=zeta; zeta_p2:=N-1 fi;
  if is(eta=0) then zeta_m:=eta; zeta_m2:=N+alpha
      else zeta_m:=zeta; zeta_m2:=0 fi;
   p:=min(i,j); q:=max(i,j);
  if is(q=i) then
      op:=[`+`,zeta_p,q-p,p,zeta_p,zeta_m2,s,r];
      R:=`-`;
      m:=p+1;
      pom:=((pR@pP-pP@@0)@@r)(pMm);
      pom:=pY(zeta_p2,zeta_m,p,`-`,pom);
      z:=zeta_p;
      zeta_p:=zeta_p2;
      zeta_p2:=z;
      R:=`+`;
      pUm:=pZ(zeta_p,q-p,`+`,pom);
      m:=q+1;
      pom:=((pR@pP+pP@@0)@@s)(pMp);
      pUp:=pW(zeta_p,zeta_m2,q-p+1,q,`+`,pom);
      stm:=max(r+p+stmm,s+p+stmp);
      stp:=max(r+q-p+stpm,s+stpp)
     else
      op:=[`-`,zeta_m,q-p,p,zeta_p,zeta_m2,s,r];
      R:=`+`;
      m:=p+1;
      pom:=((pR@pP+pP@@0)@@s)(pMp);
      pom:=pY(zeta_p,zeta_m2,p,`+`,pom);
      z:=zeta_m;
      zeta_m:=zeta_m2;
      zeta_m2:=z;
      R:=`-`;
      pUp:=pZ(zeta_m,q-p,`-`,pom);
      m:=q+1;
      pom:=((pR@pP-pP@@0)@@r)(pMm);
      pUm:=pW(zeta_p2,zeta_m,q-p+1,q,`-`,pom);
      stm:=max(r+p+stmm,s+p+stmp);
      stp:=max(s+q-p+stpp,r+stpm)
    fi;
   s0:=0;
   o0:=pUp+pUm;
    for l from -stm to stp
      do
        s0:=s0+factor(simplify(coeff(o0,b[k+l],1)))*b[k+l]
      od;
   [s0,stm,stp,op]
end:

Hahn:=proc(lm,lp)
```

```
  options `Copyright 2001 by Pawel Wozny`;
  global R;
  local Pp,Pm,np,nm,Kp,Km,m,p,i,j,n,l,oper,w,d,d2,A,B,pom,sp,sm,prze,o;
   nm:=nops(lm); np:=nops(lp);
   R:=`-`;
    if nm=0 then Pm:=0 else
      nm:=nm-1; Pm:=sum(simplify(lm[n+1])*(bD@@n)(f(x)),n=0..nm)
    fi;
   R:=`+`;
    if np=0 then Pp:=0 else
      np:=np-1; Pp:=sum(simplify(lp[l+1])*(bD@@l)(f(x)),l=0..np)
    fi;
   Km:=canonical_form(nm,Pm,`-`);
   Kp:=canonical_form(np,Pp,`+`);
   m:=integrate_canonical_form(nm,Km,`-`);
   p:=integrate_canonical_form(np,Kp,`+`);
   oper:=p_i_m(p[1],p[2],p[3],p[4],p[5],p[6],m[1],m[2],m[3],m[4],m[5],m[6]);
   sm:=0; sp:=0; d:=1; d2:=1;
    for i from -oper[2] to oper[3]
     do
       pom:=coeff(oper[1],b[k+i],1);
       B[i]:=factor(simplify(pom*h(k+i)/h(k)));
        if B[i]<>0 then
          d:=lcm(d,denom(B[i]));
          d2:=gcd(d2,numer(B[i]));
          sp:=max(sp,i);
          sm:=min(sm,i)
        fi
     od;
   prze:=-sm-floor((sp-sm)/2);
   w:=factor(simplify(d/d2));
   o:=oper[4];
    for j from sm+prze to sp+prze
     do
       A[j]:=0;
        if B[j-prze]<>0 then
          A[j]:=factor(simplify(shift(B[j-prze]*w,k,prze)));
          print(`A[`.j.`]`=A[j])
        fi
     od;
   print('w'=shift(w,k,prze));
    if o[1]=`+` then
      print('s'=prze,'zeta'=o[2],'eta'=o[6],'d'=o[3],'t'=o[4],
            'u'=o[7],'v'=o[8]);
      print(T=1/'h'(k+prze)*E^'s'*Z['zeta']^`+(d)`*Y['zeta','eta']^`+(t)`*
            Z['zeta']^`+(t)`*(D^`-`)^'u'*(D^`+`)^'v')
     else
      print('s'=prze,'epsilon'=o[2],'zeta'=o[5],'eta'=o[6],'d'=o[3],
            't'=o[4],'u'=o[7],'v'=o[8]);
      print(T=1/'h'(k+prze)*E^'s'*Z['epsilon']^`-(d)`*
            Y['zeta','eta']^`+(t)`*Z['zeta']^`+(t)`*(D^`-`)^'u'*(D^`+`)^'v')
    fi:
   [sm+prze,sp+prze,A,shift(w,k,prze)]
end:
```

B.3. **Expansions in the little q-Jacobi polynomials.**

```
##################################################################
#
# Warning!!! To run this program you need file qsum.mpl, which is available
# ~~~~~~~~~~ on http://www.imn.htwk-leipzig.de/~koepf.
```

```
#
###################################################################
#
# Constructing recurrence equation for the little q-Jacobi series
# coefficients of a solution of a linear q-difference equation.
#
###################################################################
#
# Version 1.0 for MapleV release 4
#
# Copyright (c) Pawel Wozny (Pawel.Wozny@ii.uni.wroc.pl), 07/02/2001
#
###################################################################
#
# Let f(x) be a solution of a linear q-difference equation
#
#             n                           m
#           -----                       -----
#            \              i            \               j
# (1)         )   u_i(x) (D-)  f(x)  +    )   v_j(x) (D+)  f(x) = g(x),
#            /                           /
#           -----                       -----
#           i = 0                       j = 0
#
# where u_i(x), v_j(x) are polynomials, and
#
#                                    f(q x) - f(x)
#                             (D+) = -------------,
#                                      x (q - 1)
#
#                               f(x/q) - f(x)
#                        (D-) = -------------   (0 < q <1 ).
#                                x (1/q - 1)
#
# Let a_k[f] be the Fourier coefficients of the function f(x),
#
#                                 -----
#                                  \
# (2)                      f(x) =   )   a_k[f] p_k(x),
#                                  /
#                                 -----
#                                   k
#
# where p_k(x) is the k-th monic little q-Jacobi polynomial. This program
# finds recurrence relation for a_k[f], of the form
#
#                          r + d
#                          -----
#                           \
# (3)                        )    A_i(k) a_{i+r}[f] = B(k)
#                           /
#                          -----
#                          i = -r
#
# under assumption that a_k[g] are known explicitly. Here r is a natural
# number and d  {0,1}.
#
# Main procedure
# ~~~~~~~~~~~~~~
#   - l_qJacobi(U,V)
```

```
#
# Input
# ~~~~~
#   - vectors U := [u_0(x),u_1(x),...,u_n(x)] and
#     V := [v_0(x),v_1(x),...,v_m(x)]
#
# Output
# ~~~~~~
#   - vector [-r,r+d,A], where  A[-r..r+d] is a table of A_l(k)
#     (l=-r..r+d)
#   - operator T such that  B(k)= T(a_k[g])
#     (see Algorithm 3.1, Step 6, Eq. (**)) is given on the screen
#
###################################################################
#
# Example:
# ~~~~~~~~
#  Let f(x)=x^n. This function satisfies the first-order
#  q-difference equation
#
#                      x(D+)(f(x))-[(q^n-1)/(q-1)]f(x)=0.
#
#  We are loking for the recurrence equation for a_k[f]. We should initialize
#  Maple session as follows:
#
#  >read(`l_jacobi.mpl`):
#  >l_qJacobi([],[-qnum(n),x]);
#
###################################################################

read(`qsum.mpl`):
with(LREtools):
bE:=(f,i)->subs({x=q^i*x,X=q^i*X},f):
qpochhammer:=(x,q,kk)->product(1-x*q^mm,mm=0..kk-1):
qnum:=kk->(q^kk-1)/(q-1):
eps:=()->if R=`+` then 1 else q fi:
ww:=(n,q)->if (is(n=0) or is(q=1)) then 1
            else
             if R=`+` then 1/(q^((n+1)*(n+2)/2-2*n-1)*(q-1)^n)
               else 1/(q^(-(n+1)*(n+2)/2+2*n+1)*(q^(-1)-1)^n)
             fi
           fi:
bD:=f->if R=`+` then (bE(f,1)-f)/(X*(q-1)) else (bE(f,-1)-f)/(X*(q^(-1)-1)) fi:
bU:=f->sigma(x)/eps()*bD(f,x)+tau(x)*f:
bV:=f->if R=`+` then (x-zeta_p)*bD(f,x) else (x-zeta_m)/q*bD(f) fi:
zeta_p2:=0:  zeta_p:=1/(b*q):  zeta_m2:=1:  zeta_m:=0:
ni_p:=(zp2,zm2)->qsimplify(2*tau(zm2)/(2*a*b*q^2*(zm2-zp2))):
ni_m:=(zp2,zm2)->qsimplify(2*tau(zp2)/(2*(zp2-zm2))):
gamma_p:=(zp2,zm2,k)->qsimplify(-q^(k-1)*lambda(k)/RR(k-1)+ni_p(zp2,zm2)):
gamma_m:=(zp2,zm2,k)->qsimplify(1/(q-1-q^(1-k)*RR(k-1)/lambda(k))+ni_m(zp2,zm2)):
sigma:=x->if R=`+` then a*q*x*(b*q*x-1) else x*(x-1) fi:
sigmam:=x->x*(x-1):
tau:=x->(a*q*(b*q*x-1)+1-x)/(q-1):
lambda:=k->-(q^k-1)*(a*b*q^(k+1)-1)/(q^(k-1)*(q-1)^2):
v_p:=k->qsimplify(-q^(k-1)*lambda(k)/RR(k-1)):
v_m:=k->qsimplify(1/(q-1-q^(1-k)*RR(k-1)/lambda(k))):
t_1:=diff(tau(x),x):  t_0:=tau(0):
s_2:=simplify(diff(sigmam(x),x,x)/2+(q-1)*t_1):
s_1:=simplify(subs(x=0,diff(sigmam(x),x))+(q-1)*tau(0)):
s_0:=sigmam(0):
```

```
RR:=k->s_2*qnum(k)+t_1:
h:=n->if n=0 then 1 elif n>0 then product(ksi0(k+ii),ii=1..n)
         else product(1/ksi0(k+ii),ii=n+1..0) fi:
ksi0:=k->a*q^(2*k)*(q^k-1)*(a*q^k-1)*(b*q^k-1)*(a*b*q^k-1)
          /((a*b*q^(2*k)-1)^2*(a*b*q^(2*k)-q)*(a*b*q^(2*k+1)-1)):
ksi1:=k->q^k*(1+a+a*(1+a)*b*q^(2*k+1)-a*(1+b)*q^k*(1+q))
          /((a*b*q^(2*k)-1)*(a*b*q^(2*k+2)-1)):
ksi2:=k->1:
delta_p0:=k->qsimplify(-q^(1-k)*RR(k-1)*ksi0(k)/lambda(k)):
delta_p1:=k->qsimplify(-q*t_0/((q+1)*lambda(k))-
                      q*ksi1(k)*((q-1)*qnum(-k)*RR(k-1)+t_1)/
                      ((q+1)*lambda(k))):
delta_p2:=k->qsimplify(s_2*qnum(k)/lambda(k)):
delta_m0:=k->qsimplify(delta_p0(k)+(q-1)*ksi0(k)):
delta_m1:=k->qsimplify(delta_p1(k)+(q-1)*ksi1(k)):
delta_m2:=k->qsimplify(delta_p2(k)+(q-1)*ksi2(k)):
pX:=f->ksi0(k)*shift(f,k,-1)+ksi1(k)*f+ksi2(k)*shift(f,k,1):
pD:=f->if R=`+` then
         delta_p0(k)*shift(f,k,-1)+delta_p1(k)*f+delta_p2(k)*shift(f,k,1):
        else
         delta_m0(k)*shift(f,k,-1)+delta_m1(k)*f+delta_m2(k)*shift(f,k,1):
       fi:
lambda_pD:=f->if R=`+` then
                -lambda(k)/q*(delta_m0(k)*shift(f,k,-1)+
                 delta_m1(k)*f+delta_m2(k)*shift(f,k,1))
               else
                -lambda(k)*(delta_p0(k)*shift(f,k,-1)+
                 delta_p1(k)*f+delta_p2(k)*shift(f,k,1))
              fi:
pDelta:=f->if R=`+` then
             if m>0 then (pP@pR)(f)+(q-1)*(pQ@pR)(f)+zeta_p*(q-1)*f else
               delta_m0(k)*shift(f,k,-1)+delta_m1(k)*f+
               delta_m2(k)*shift(f,k,1) fi
            else
             if m>0 then (pP@pR)(f)-(q-1)*(pQ@pR)(f)-zeta_m*(q-1)*f else
               delta_p0(k)*shift(f,k,-1)+delta_p1(k)*f+
               delta_p2(k)*shift(f,k,1) fi
            fi:
pP:=f->if R=`+` then
         f+
         b*q^m*qpochhammer(a*b*q^(2*k+1),q,3)*(a*b*q^(2*k+2)-1)/
            ((b*q^(k+1)-b*q*zeta_p2)*(1-zeta_p*a*b*q^(k+2))*
            (a*b*q^(k+1)-1)*qpochhammer(a*b*q^(2*k+m+1),q,2))*shift(f,k,1)
        else
         f+
         q^(m-1)*qpochhammer(a*b*q^(2*k+1),q,3)*(a*b*q^(2*k+2)-1)
            /(q^k*(a*q^(k+1)-zeta_m)*
            (zeta_m2*b*q^(k+1)-1)*(a*b*q^(k+1)-1)*
            qpochhammer(a*b*q^(2*k+m+1),q,2))*shift(f,k,1)
       fi:
pQ:=f->if R=`+` then
         v_p(k)*f+
         qpochhammer(a*b*q^(2*k+1),q,3)*(a*b*q^(2*k+2)-1)*(1-a*b*q^(k+m+1))/
            (a*q^(k+1)*(q-1)*(b*q^(k+1)-b*q*zeta_p2)*(1-zeta_p*a*b*q^(k+2))*
            (a*b*q^(k+1)-1)*qpochhammer(a*b*q^(2*k+m+1),q,2))*shift(f,k,1)
        else
         v_m(k)*f+
         q^(m-1)*qpochhammer(a*b*q^(2*k+1),q,3)*(a*b*q^(2*k+2)-1)*
            (1-a*b*q^(k+m+1))/(q^k*(q-1)*(a*q^(k+1)-zeta_m)*
            (zeta_m2*b*q^(k+1)-1)*(a*b*q^(k+1)-1)*
```

```
            qpochhammer(a*b*q^(2*k+m+1),q,2))*shift(f,k,1)
       fi:
pR:=f->if R=`+` then
         delta_p0(k)*shift(f,k,-1)+
         a*q^k*(q-1)*(b*q^(k+m)-b*q*zeta_p2)*(1-zeta_p*a*b*q^(k+m+1))/
            qpochhammer(a*b*q^(2*k+m),q,2)*f
        else
         delta_m0(k)*shift(f,k,-1)+
         q^k*(q-1)*(a*q^(k+m)-zeta_m)*(zeta_m2*b*q^(k+m)-1)/
            qpochhammer(a*b*q^(2*k+m),q,2)*f
       fi:
pF:=f->if R=`+` then
         ni_p(zeta_p2,zeta_m2)*delta_m0(k)/gamma_p(zeta_p2,zeta_m2,k-1)*
         shift(f,k,-1)+
         b*q^(k+1)*(q-1)*(a*q-zeta_m)*(1-zeta_p)*
            (zeta_m2*q^(k+m-1)-zeta_p2)/qpochhammer(a*b*q^(2*k+m),q,2)*f
        else
         q*ni_m(zeta_p2,zeta_m2)*delta_p0(k)/gamma_m(zeta_p2,zeta_m2,k-1)*
         shift(f,k,-1)
         -a*q^(2*k+m)*(b*q-zeta_m)*(q-1)*
            (zeta_m2*q^(k+m-1)-zeta_p2+zeta_m*(1+zeta_p2))*(a*b*q^2*zeta_p-1)/
            qpochhammer(a*b*q^(2*k+m),q,2)*f
       fi:

find_bQi:=proc(i,w,r)
  global R;
  local test1,test2,j,wx,wx2,wx3,qq,stala,aa,bb,z1,z2,s,m;
   R:=r;
    if qsimplify(w)=0 then
      RETURN([0,0,0,0,0,r])
    fi;
   test1:=true; test2:=true; aa:=0; bb:=0;
   wx:=unapply(qsimplify(w),x);
   wx2:=unapply(qsimplify(w),x);
   wx3:=unapply(qsimplify(w),x);
    if r=`+` then
      z1:=0; z2:=1/(b*q); s:=-1; m:=a*b*q^2/eps()
     else
      z1:=1; z2:=0; s:=1; m:=1/eps()
    fi;
    for j from 1 to i while (test1 or test2)
     do
        if (test1 and simplify(wx2(z1*q^(s*(i-j))))=0) then
          aa:=aa+1; wx2:=unapply(qsimplify(wx2(x)/(x*q^(-s*(i-j))-z1)),x)
           else test1:=false fi;

        if (test2 and simplify(wx3(z2*q^(s*(i-j))))=0) then
          bb:=bb+1; wx3:=unapply(qsimplify(wx3(x)/(x*q^(-s*(i-j))-z2)),x)
           else test2:=false fi
     od;
   qq:=simplify(wx(x)/(product(x*q^(s*(k-i))-z1,k=1..aa)*
                      product(x*q^(s*(l-i))-z2,l=1..bb)));
    if min(aa,bb)=bb then
      stala:=ww(i-aa,q)*ww(aa-bb,q)*ww(bb,q)*q^(s*((i-aa)*(aa-bb)+bb*(i-bb)));
      [i-aa,aa-bb,z1,bb,factor(qsimplify(eps()^(aa-bb)/(stala*m^bb)*
       subs(X=x,bE(qq*X^i,s*i))))),r]
     else
      stala:=ww(i-bb,q)*ww(bb-aa,q)*ww(aa,q)*q^(s*((i-bb)*(bb-aa)+aa*(i-aa)));
      [i-bb,bb-aa,z2,aa,factor(qsimplify(eps()^(bb-aa)/(stala*m^aa)*
        subs(X=x,bE(qq*X^i,s*i))))),r]
```

```
    fi
end:

canonical_form:=proc(n,p,r)
  global zeta_p,zeta_m,R;
  local i,j,k,w,bQ,qq,s,P;
   R:=r;
   P[n]:=p;
    if R=`+` then
      s:=1 else s:=-1 fi;
    for k from 0 to n
     do
       w[k]:=coeff(P[n],f(x*q^(s*k)),1)
     od;
   bQ:=array(0..n);
    for i from n to 1 by -1
     do
       qq:=find_bQi(i,w[i],r);
       bQ[i]:=qq;
        if r=`+` then zeta_p:=qq[3] else zeta_m:=qq[3] fi;
       P[i-1]:=P[i]-(bD@@qq[1]@bV@@qq[2]@bU@@qq[4])(qq[5]*f(x));
        for j from 0 to i-1
         do
           w[j]:=coeff(P[i-1],f(x*q^(s*j)),1)
         od
     od;
    bQ[0]:=[0,0,0,0,factor(qsimplify(subs(X=x,w[0]))),r];
    bQ
end:

poly_to_oper:=proc(w,f)
  local i,p,j;
    for i from 0 to degree(w,x)
     do
       p[i]:=coeff(w,x,i)*(pX@@i)(f)
     od;
   sum(p[j],j=0..degree(w,x))
end:

pS:=proc(z,i,j,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,S;
   R:=r;
   S:=(pP@@0)(f);
    if r=`+` then
       if is(z=0) then zeta_p:=z; zeta_p2:=1/(b*q)
        else zeta_p:=z; zeta_p2:=0 fi
     else
      if is(z=1) then zeta_m:=z; zeta_m2:=0
       else zeta_m:=z; zeta_m2:=1 fi
    fi;
    for l from j to i
     do
       m:=l;
       S:=pP(S)
     od;
   S
end:

pZ:=proc(z,n,r,f)
```

```
   pS(z,n,1,r,f)
end:

pU:=proc(z,n,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,Q;
   R:=r;
   Q:=(pQ@@0)(f);
    if r=`+` then
       if is(z=0) then zeta_p:=z; zeta_p2:=1/(b*q)
        else zeta_p:=z; zeta_p2:=0 fi
     else
      if is(z=1) then zeta_m:=z; zeta_m2:=0
       else zeta_m:=z; zeta_m2:=1 fi
    fi;
    for l from 1 to n
     do
       m:=l;
       Q:=pQ(Q)
     od;
   Q
end:

pM:=proc(z,i,j,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,_R;
   R:=r;
   _R:=(pR@@0)(f);
    if r=`+` then
       if is(z=0) then zeta_p:=z; zeta_p2:=1/(b*q)
        else zeta_p:=z; zeta_p2:=0 fi
     else
      if is(z=1) then zeta_m:=z; zeta_m2:=0
       else zeta_m:=z; zeta_m2:=1 fi
    fi;
    for l from j to i by -1
     do
       m:=l;
       _R:=pR(_R)
     od;
   _R
end:

pN:=proc(z,n,r,f)
   pM(z,1,n,r,f)
end:

pW:=proc(zp,zm,i,j,r,f)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local l,V;
   R:=r;
   V:=(pF@@0)(f);
   if is(zp=0) then zeta_p:=zp; zeta_p2:=1/(b*q)
     else zeta_p:=zp; zeta_p2:=0 fi;
   if is(zm=0) then zeta_m:=zm; zeta_m2:=1
     else zeta_m:=zm; zeta_m2:=0 fi;
    for l from j to i by -1
     do
       m:=l;
       V:=pF(V)
```

```
     od;
   V
end:

pY:=proc(zp,zm,n,r,f)
   pW(zp,zm,1,n,r,f)
end:

integrate_canonical_form:=proc(n,Q,r)
  global R;
  local Omega,z,z2,i,j,l,p,p2,qq,q2,e,epsilon,epsilon2,
        pom,O,sO,oO,stp,stm,sm,sp,t;
  R:=r;
    if r=`+` then
      z:=0; z2:=1/(b*q)
     else
      z:=1; z2:=0
    fi;
   Omega[z]:=[]; Omega[z2]:=[]; stp:=0; stm:=0;
   p[z]:=0; p[z2]:=0; qq[z]:=0; qq[z2]:=0;
    for i from 0 to n
     do
        if is(Q[i][3]=z) then
          Omega[z]:=[op(Omega[z]),i];
          p[z]:=max(p[z],Q[i][1]);
          qq[z]:=max(qq[z],Q[i][1]+Q[i][2]);
        else
          Omega[z2]:=[op(Omega[z2]),i];
          p[z2]:=max(p[z2],Q[i][1]);
          qq[z2]:=max(qq[z2],Q[i][1]+Q[i][2])
        fi
     od;
   qq[z]:=qq[z]-p[z];
   qq[z2]:=qq[z2]-p[z2];
    if is(p[z]+qq[z]<p[z2]+qq[z2]) then epsilon:=z2; epsilon2:=z
     else epsilon:=z; epsilon2:=z2 fi;
   p2:=max(p[epsilon],p[epsilon2]+qq[epsilon2]);
   q2:=p[epsilon]+qq[epsilon]-p2;
    for e in [epsilon,epsilon2]
     do
      for j in Omega[e]
       do
         pom:=eps()^Q[j][1]*(lambda_pD@@Q[j][4])(poly_to_oper(Q[j][5],b[k]));
         pom:=pU(Q[j][3],Q[j][2],r,pom);
         pom:=pM(e,qq[e]+1,qq[e]+p[e]-Q[j][1],r,
                 pS(e,qq[e]+p[e]-Q[j][1],Q[j][2]+1,r,pom));
         sp:=degree(Q[j][5],x)+Q[j][4]+qq[e]+p[e]-Q[j][1];
         sm:=degree(Q[j][5],x)+Q[j][4]+p[e]-Q[j][1];
          if is(e=epsilon) then
            O[j]:=pM(e,q2+1,qq[e],r,pom);
            sm:=sm+qq[e]-q2
           else
            O[j]:=pZ(epsilon,q2,r,pN(e,p2-p[e],r,
                     pS(e,p2-p[e],qq[e]+1,r,pom)));
            sp:=sp+p2+q2-p[e]-qq[e];
            sm:=sm+p2-p[e]
          fi;
        stp:=max(stp,sp);
        stm:=max(stm,sm)
       od
```

```
     od;
   s0:=0;
   o0:=sum(O[t],t=0..n);
    for l from -stm to stp
      do
        s0:=s0+factor(qsimplify(coeff(o0,b[k+l],1)))*b[k+l]
      od;
   [epsilon,q2,p2,s0,stm,stp]
end:

p_i_m:=proc(zeta,_i,_r,_pMp,_stmp,_stpp,eta,_j,_s,_pMm,_stmm,_stpm)
  global R,m,zeta_p,zeta_p2,zeta_m,zeta_m2;
  local p,qq,pom,pUp,pUm,z,s0,o0,stm,stp,l,
        i,r,pMp,stmp,stpp,j,s,pMm,stmm,stpm,op;
   i:=_i;r:=_r;pMp:=_pMp;stmp:=_stmp;stpp:=_stpp;
   j:=_j;s:=_s;pMm:=_pMm;stmm:=_stmm;stpm:=_stpm;
    if is(zeta=0) then zeta_p:=zeta; zeta_p2:=1/(b*q)
      else zeta_p:=zeta; zeta_p2:=0 fi;
    if is(eta=0) then zeta_m:=eta; zeta_m2:=1
      else zeta_m:=eta; zeta_m2:=0 fi;
    if (i>0 and j>0 and (zeta_p-zeta_m2)*(zeta_m-zeta_p2)=0) then
      if i<j then
        r:=r+i; pMp:=pN(zeta,i,`+`,pMp); stpm:=stpm+i; i:=0;
       else
        s:=s+j; pMm:=pN(eta,j,`-`,pMm); stmm:=stmm+i; j:=0;
      fi
    fi;
   p:=min(i,j); qq:=max(i,j);
    if is(qq=i) then
      op:=[`+`,zeta_p,qq-p,p,zeta_p,zeta_m2,s,r];
      R:=`-`;
      m:=p;
      pom:=((pDelta)@@r)(pMm);
      pom:=pY(zeta_p2,zeta_m,p,`-`,pom);
      z:=zeta_p;
      zeta_p:=zeta_p2;
      zeta_p2:=z;
      R:=`+`;
      pUm:=q^(r*s)*pZ(zeta_p,qq-p,`+`,pom);
      m:=qq;
      pom:=((pDelta)@@s)(pMp);
      pUp:=pW(zeta_p,zeta_m2,qq-p+1,qq,`+`,pom);
      stm:=max(r+p+stmm,s+p+stmp);
      stp:=max(r+qq-p+stpm,s+stpp)
     else
      op:=[`-`,zeta_m,qq-p,p,zeta_p,zeta_m2,s,r];
      R:=`+`;
      m:=p;
      pom:=((pDelta)@@s)(pMp);
      pom:=pY(zeta_p,zeta_m2,p,`+`,pom);
      z:=zeta_m;
      zeta_m:=zeta_m2;
      zeta_m2:=z;
      R:=`-`;
      pUp:=pZ(zeta_m,qq-p,`-`,pom);
      m:=qq;
      pom:=((pDelta)@@r)(pMm);
      pUm:=q^(r*s)*pW(zeta_p2,zeta_m,qq-p+1,qq,`-`,pom);
      stm:=max(r+p+stmm,s+p+stmp);
      stp:=max(s+qq-p+stpp,r+stpm)
```

```
    fi;
   s0:=0;
   o0:=pUp+pUm;
    for l from -stm to stp
      do
        s0:=s0+factor(qsimplify(coeff(o0,b[k+l],1)))*b[k+l]
      od;
   [s0,stm,stp,op]
end:

l_qJacobi:=proc(lm,lp)
  options `Copyright 2001 by Pawel Wozny`;
  global R;
  local Pp,Pm,np,nm,Kp,Km,m,n,l,p,i,j,oper,d,A,B,pom,sp,sm,prze,o;
  nm:=nops(lm); np:=nops(lp);
  R:=`-`;
  if nm=0 then Pm:=0 else
      nm:=nm-1; Pm:=sum(simplify(lm[n+1])*(bD@@n)(f(x)),n=0..nm)
    fi;
   R:=`+`;
    if np=0 then Pp:=0 else
      np:=np-1; Pp:=sum(simplify(lp[l+1])*(bD@@l)(f(x)),l=0..np)
    fi;
   Km:=canonical_form(nm,Pm,`-`);
   Kp:=canonical_form(np,Pp,`+`);
   m:=integrate_canonical_form(nm,Km,`-`);
   p:=integrate_canonical_form(np,Kp,`+`);
   oper:=p_i_m(p[1],p[2],p[3],p[4],p[5],p[6],m[1],m[2],m[3],m[4],m[5],m[6]);
   sm:=0; sp:=0; d:=1;
    for i from -oper[2] to oper[3]
     do
       B[i]:=qsimplify(coeff(oper[1],b[k+i],1)*(h(i)));
        if B[i]<>0 then
          sp:=max(sp,i);
          sm:=min(sm,i)
        fi
     od;
   prze:=-sm-floor((sp-sm)/2);
   o:=oper[4];
    for j from sm+prze to sp+prze
     do
       A[j]:=0;
        if B[j-prze]<>0 then
          A[j]:=factor(qsimplify(shift(B[j-prze],k,prze)));
          print(`A[`.j.`]`=A[j])
        fi
     od;
    if o[1]=`+` then
      print('s'=prze,'zeta'=o[2],'eta'=o[6],'d'=o[3],'t'=o[4],
            'u'=o[7],'v'=o[8]);
      print(T=1/'h'(k+prze)*E^'s'*Z['zeta']^`+(d)`*Y['zeta','eta']^`+(t)`*
            Z['zeta']^`+(t)`*(D^`-`)^'u'*(D^`+`)^'v');
     else
      print('s'=prze,'epsilon'=o[2],'zeta'=o[5],'eta'=o[6],'d'=o[3],
            't'=o[4],'u'=o[7],'v'=o[8]);
      print(T=1/'h'(k+prze)*E^'s'*Z['epsilon']^`-(d)`*Y['zeta','eta']^`+(t)`*
            Z['zeta']^`+(t)`*(D^`-`)^'u'*(D^`+`)^'v');
    fi:
   [sm+prze,sp+prze,A]
end:
```

References

[1] R. Álvarez-Nodarse, J. Arvesú and R. J. Yáñez, On the connection and linearization problem for discrete hypergeometric q-polynomials, *J. Math. Anal. Appl.* **257**, 52–78.

[2] R. Álvarez-Nodarse, N. R. Quintero and A. Ronveaux, On the linearization problem involving Pochhammer symbols and their q-analogues, *J. Comput. Appl. Math.* **107** (1999), 133-146.

[3] R. Álvarez-Nodarse and A. Ronveaux, Recurrence relations for connection coefficients between q-orthogonal polynomials of discrete variables in the non-uniform lattice $x(s) = q^{2s}$, *J. Phys. A: Math. Gen.* **29** (1996), 7165–7175.

[4] R. Álvarez-Nodarse, R. Yáñez and J. S. Dehesa, Modified Clebsch-Gordan-type expansions for products of discrete hypergeometric polynomials, *J. Comput. Appl. Math.* **89** (1998), 171–197.

[5] I. Area, E. Godoy, A. Ronveaux and A. Zarzo, Inversion problems in the q-Hahn tableau, *J. Symbolic Comp.* **28** (1999), 767–776.

[6] P. L. Artés, J. S. Dehesa, A. Martinez-Finkelstein and J. Sánchez-Ruiz, Linearization and connection coefficients for hypergeometric-type polynomials, *J. Comput. Appl. Math.* **99** (1998), 15–26.

[7] R. Askey, *Orthogonal Polynomials and Special Functions*, Regional Conf. Ser. Appl. Math. **21** (SIAM, Philadelphia, PA, 1975).

[8] T. S. Chihara, *An Introduction to Orthogonal Polynomials* (Gordon and Breach, New York, 1978).

[9] M. Foupouagnigni, A. Ronveaux and W. Koepf, The fourth-order q-difference equation satisfied by the first associated q-classical orthogonal polynomials, *J. Comput. Appl. Math.* **101** (1999), 231–236.

[10] G. Gasper, Projection formulas for orthogonal polynomials of a discrete variable, *J. Math. Anal. Appl.* **45** (1974), 176–198.

[11] G. Gasper and M. Rahman, *Basic hypergeometric series* (Cambridge University Press, Cambridge 1990).

[12] E. Godoy, A. Ronveaux, A. Zarzo and I. Area, Minimal recurrence relations for connection coefficients between classical orthogonal polynomials: continuous case, *J. Comput. Appl. Math.* **84** (1997), 257–275.

[13] W. Hahn, Über Orthogonalpolynome, die q-Differenzengleichungen genügen, *Math. Nachr.* **2** (1949), 4–34.

[14] M. N. Hounkonnou, S. Belmehdi and A. Ronveaux, Linearization of arbitrary product of classical orthogonal polynomials, *Appl. Math.* **27** (2000), 187–196.

[15] R. Koekoek and R.F. Swarttouw, *The Askey scheme of hypergeometric orthogonal polynomials and its q-analogue*, Fac. Techn. Math. Informatics, Delft Univ. of Technology, Rep. 98-17, Delft (1998).

[16] T. Koornwinder, Compact quantum groups and q-special functions, in: *Representations of Lie groups and quantum groups*, eds. V. Baldoni and M.A. Picardello, Pitman Research Notes in Mathematics Series, vol. 311 (Longman, New York 1994), pp. 46–128.

[17] W. Koepf and D. Schmersau, Algorithms for classical orthogonal polynomials, Konrad-Zuse-Zentrum Berlin, Preprint SC 96-23, September 1996.

[18] W. Koepf and D. Schmersau, Representations of orthogonal polynomials, *J. Comput. Appl. Math.* **90**, 57–94.

[19] S. Lewanowicz, A new approach to the problem of constructing recurrence relations for the Jacobi coefficients, *Zastos. Mat.* **21** (1991) 303–326.

[20] S. Lewanowicz, Results on the associated Jacobi and Gegenbauer polynomials, *J. Comput. Appl. Math.* **49** (1993) 137–143.

[21] S. Lewanowicz, Second-order recurrence relation for the linearization coefficients of the classical orthogonal polynomials, *J. Comput. Appl. Math.* **69** (1996) 159–170.

[22] S. Lewanowicz, Results on the associated classical orthogonal polynomials, *J. Comput. Appl. Math.* **65** (1995), 215–231.

[23] S. Lewanowicz, Recurrence relations for the connection coefficients of orthogonal polynomials of a discrete variable, *J. Comput. Appl. Math.* **76** (1996), 213–229.

[24] S. Lewanowicz, Recurrence relations for the connection coefficients of orthogonal polynomials of a discrete variable on the lattice $x(s) = q^{2s}$, *J. Comput. Appl. Math.* **99** (1998), 275–286.

[25] S. Lewanowicz, Recursion formulae for basic hypergeometric functions, in *Numerical Analysis in the 20th Century, Vol. I: Approximation Theory*, eds. J. Wimp & L. Wuytack, *J. Comput. Appl. Math.* **121** (2000), 297–312.

[26] S. Lewanowicz, Recurrences for the coefficients of series expansions with respect to classical orthogonal polynomials, *Appl. Math.* **29** (2002), 97–116.

[27] S. Lewanowicz, Representations for the first associated q-classical orthogonal polynomials, *J. Comput. Appl. Math.* **150** (2003), 311-327.

[28] S. Lewanowicz, Construction of recurrences for the coefficients of series expansions in q-classical orthogonal polynomials, *J. Comput. Appl. Math.* **153** (2003), 295–309.

[29] S. Lewanowicz, E. Godoy, I. Area, A. Ronveaux and A. Zarzo, Recurrence relations for the coefficients of the Fourier series expansions with respect to q-classical orthogonal polynomials, *Numerical Algorithms* **23** (2000), 31–50.

[30] Y. L. Luke, *The special functions and their approximations*, 2 vols. (Academic Press, New York 1969).

[31] J. C. Medem, *Polinomios ortogonales q-semiclásicos*, PhD dissertation, Universidad Politécnica de Madrid, 1996.

[32] J. C. Medem, R. Álvarez-Nodarse and F. Marcellán, On the q-classical polynomials: A distributional study, *J. Comput. Appl. Math.* **135** (2001), 157–196.

[33] A. F. Nikiforov and V. B. Uvarov, *Special Functions of Mathematical Physics* (Birkhäuser, Basel, 1988).

[34] A. F. Nikiforov, S. K. Suslov and V. B. Uvarov, *Classical Orthogonal Polynomials of a Discrete Variable*, Springer-Verlag, Berlin, 1991.

[35] A. Ronveaux, Orthogonal polynomials: connection and linearization coefficients, in: M. Alfaro et al. (Eds), Proc. *Internat. Workshop Orthogonal Polynomials in Math. Phys.*, Leganés, 1996, pp. 131–142.

[36] A. Ronveaux, I. Area, E. Godoy and A. Zarzo, Lectures on recursive approach to connection and linearization coefficients between polynomials. Introduction, in Proc. *Workshop on Special Functions and Differential Equations*, Madras, Jan. 13–24, 1997.

[37] A. Ronveaux, S. Belmehdi, E. Godoy and A. Zarzo, Recurrence relations approach for connection coefficients – Applications to classical discrete orthogonal polynomials, in *Symmetries and Integrability of Difference Equations*, eds. D. Levi, L. Vinet and P. Winternitz, CRM Proceedings and Lecture Notes Series, vol. 9 (1996), AMS, Providence (RI), pp. 321–337.

[38] A. Ronveaux, S. Houkonnou and S. Belmehdi, Generalized linearization problems, *J. Phys. A: Math. Gen.* **28** (1995) 4423–4430

[39] A. Ronveaux, A. Zarzo I. Area and E. Godoy, Bernstein bases and Hahn-Eberlein orthogonal polynomials, *Int. Transf. Spec. Func.* **7** (1998), 87-96.

[40] A. Ronveaux, A. Zarzo I. Area and E. Godoy, Classical orthogonal polynomials: Dependence of parameters, *J. Comput. Appl. Math.* **121** (2000), 95–112.

[41] A. Ronveaux, A. Zarzo and E. Godoy, Recurrence relation for connection coefficients between two families of orthogonal polynomials, *J. Comput. Appl. Math.* **62** (1995) 67–73.

[42] J. Sánchez-Ruiz, P. Artés, A. Martínez-Finkelshtein and J. S. Dehesa, General linearization formulae for products of continuous hypergeometric-type polynomials, *J. Phys. A: Math. Gen.* **32** (1999), 1–22.

[43] J. Sánchez-Ruiz and J. S. Dehesa, Expansions in series of orthogonal hypergeometric polynomials, *J. Comput. Appl. Math.* **89** (1998), 155–170.

[44] R. Szwarc, Linearization and connection coefficients of orthogonal polynomials, *Mh. Math.* **113** (1992), 319–329.

[45] R. Szwarc, Connection coefficients of orthogonal polynomials, *Canad. Math. Bull.* **35** (1992), 548–556.

[46] R. Szwarc, Nonnegative linearization and quadratic transformation of Askey-Wilson polynomials, *Canad. Math. Bull.* **39** (1996), 241–249.

[47] J. Wimp, *Computation with recurrence relations* (Pitman, Boston 1984).

[48] P. Woźny, Recurrence relations for the coefficients of expansions in classical orthogonal polynomials of a discrete variable *Appl. Math.* **30** (2003), 89–107.

[49] A. Zarzo, I. Area, E. Godoy and A. Ronveaux, Results for some inversion problems for classical continuous and discrete orthogonal polynomials, *J. Phys. A: Math. Gen.* **30** (1997) L35–L40.

E-mail address: {Stanislaw.Lewanowicz,Pawel.Wozny}@ii.uni.wroc.pl

INSTITUTE OF COMPUTER SCIENCE, UNIVERSITY OF WROCŁAW, 51-151 WROCŁAW, POLAND