Algorithmic Determination of q-Power Series for q-Holonomic Functions

Torsten Sprenger, Wolfram Koepf

Institute of Mathematics, University of Kassel, D-34132 Kassel, Germany

Abstract

In [Koepf (1992)] it was shown how for a given holonomic function a representation as a formal power series of hypergeometric type can be determined algorithmically. This algorithm – that we call FPS algorithm (Formal Power Series) – combines three steps to obtain the desired representation. The authors implemented this algorithm in the computer algebra system *Maple* as `` `convert/FormalPowerSeries` `` which is always successful if the input function is a linear combination of hypergeometric power series.

In this paper we give a q-analogue of the FPS algorithm for q-holonomic functions and extend this algorithm in such a way that it identifies and returns linear combinations of q-hypergeometric series. The algorithm is a combination of mainly three subalgorithms, which make use of existing algorithms from [Abramov, Paule, and Petkovšek (1998)], [Böing and Koepf (1999)] and [Abramov, Petkovšek, and Ryabenko (2000)]. We introduce two different polynomial bases for the representation of q-series and realize that they are sufficient to obtain all well-known q-hypergeometric representations of the classical q-orthogonal polynomials of the q-Hahn class [Koekoek and Swarttouw (1998)]. Then we develop an algorithm which converts a q-holonomic recurrence equation of a q-hypergeometric series with nontrivial expansion point into the corresponding q-holonomic recurrence equation for the coefficients. Furthermore, we show how the inverse problem can be handled. The latter algorithm is used to detect q-holonomic recurrences for some types of generalized q-hypergeometric functions. We implemented all presented algorithms (and many others) in *Maple* and make them available as *Maple* package `qFPS` which will be described briefly. Additionally, in some examples we show how `qFPS` can be applied to deduce special function identities in a simple way based on techniques used in [Zeilberger (1990)].

Key words: q-calculus, q-hypergeometric series, q-holonomic functions

Email addresses: sprenger@math.uni-kassel.de (Torsten Sprenger), koepf@math.uni-kassel.de (Wolfram Koepf).

URLs: www.mathematik.uni-kassel.de/~sprenger (Torsten Sprenger), www.mathematik.uni-kassel.de/~koepf (Wolfram Koepf).

1. Introduction

Assuming $\mathbb{K}$ is a field of characteristic 0 and q a variable, we define $\mathbb{F} := \mathbb{K}(q)$. The field $\mathbb{F}$ is the basic field considered. Given a function term $f(x)$, we deal with two linear q-operators, the *Hahn operator* D_q and the *q-shift operator* ε_q which are defined as follows

$$D_q f(x) := \frac{f(x) - f(qx)}{(1-q)x} \quad \text{and} \quad \varepsilon_q f(x) := f(qx).$$

We name $D_q f(x)$ the *q-derivative* and $\varepsilon_q f(x)$ the *q-shift* of $f(x)$. These operators are both linear and satisfy the product rules

$$D_q\left(f(x)\cdot g(x)\right) = f(qx) D_q g(x) + g(x) D_q f(x) \tag{1}$$
$$D_q\left(f(x)\cdot g(x)\right) = f(x) D_q g(x) + g(qx) D_q f(x) \tag{2}$$
$$\varepsilon_q\left(f(x)\cdot g(x)\right) = \varepsilon_q f(x) \cdot \varepsilon_q g(x). \tag{3}$$

A function term $f(x)$ is called *q-hypergeometric term*, if

$$r(x) := \frac{f(qx)}{f(x)} \in \mathbb{F}(x).$$

The rational function $r(x)$ is the *q-certificate* of $f(x)$. A function term $f(x)$ is an *m-fold q-hypergeometric term*, if

$$\frac{f(q^m x)}{f(x)} \in \mathbb{F}(x).$$

We call a sequence c_j a *q-hypergeometric term*, if

$$\frac{c_{j+1}}{c_j} \in \mathbb{F}(q^j).$$

This is consistent with the above definition, if we consider x on a linear q-lattice ($x = q^j$ and $c_j = f(q^j)$). We say that a series $\sum_j c_j x^j$ is *q-hypergeometric* or a *q-series*, if the coefficient c_j is a q-hypergeometric term. Furthermore, we consider the *generalized q-hypergeometric function* defined as

$${}_r\phi_s\left(\begin{matrix} a_1, \ldots, a_r \\ b_1, \ldots, b_s \end{matrix} \middle| q; x\right) := \sum_{j=0}^{\infty} \frac{(a_1;q)_j \cdots (a_r;q)_j}{(b_1;q)_j \cdots (b_s;q)_j} \frac{x^j}{(q;q)_j} \left((-1)^j q^{\binom{j}{2}}\right)^{1+s-r},$$

where

$$(a;q)_j := (1-a)\cdot(1-aq)\cdots(1-aq^{j-1})$$

is the *q-Pochhammer symbol* and $a_i, b_k \in \mathbb{F}$ [Gasper and Rahman (1990)]. More well-known q-functions are the q-analogue of a nonnegative integer j, called the *basic number*

$$[j]_q := \frac{1-q^j}{1-q} = 1 + q + q^2 + \ldots + q^{j-1},$$

the *q-factorial*

$$[j]_q! := [j]_q \cdot [j-1]_q \cdots [1]_q$$

and the *q-binomial coefficient*

$$\begin{bmatrix} n \\ j \end{bmatrix}_q := \frac{[n]_q!}{[j]_q!\,[n-j]_q!} = \frac{(q;q)_n}{(q;q)_j\,(q;q)_{n-j}}$$

which are all q-hypergeometric terms w.r.t. j and n. Obviously, the generalized q-hypergeometric function is indeed a q-hypergeometric series.

In this paper we present the q-FPS algorithm, which determines a linear combination of q-hypergeometric series for a given q-holonomic function $f(x)$ (this type of functions will be discussed in the forthcoming section in detail). The algorithm consists of three steps, which will be explained in Section 3, 4 and 5 respectively. Here is a rough overview of the three main parts of the algorithm.

(1) Determine a q-holonomic recurrence equation (or differential equation) for the given q-holonomic function $f(x)$.
(2) Convert this q-holonomic recurrence equation (or differential equation) into a q-recurrence equation for the coefficients of the q-power series of $f(x)$.
(3) Determine all q-hypergeometric solutions of the latter q-recurrence equation.

2. q-Holonomic Functions

The first step of the q-FPS algorithm is to determine a *q-holonomic differential equation* or *q-holonomic recurrence equation* for $f(x)$, that is a homogeneous q-differential or q-recurrence equation

$$\sum_{k=0}^{n} a_k(x) D_q^k f(x) = 0 \qquad \text{or} \qquad \sum_{k=0}^{n} b_k(x) \varepsilon_q^k f(x) = 0$$

which is linear and has polynomial coefficients, e. g. $a_k, b_k \in \mathbb{F}[x]$. In q-calculus every q-differential equation corresponds to a q-recurrence equation, as there exist the following explicit relationships between higher q-derivatives and higher q-shifts [Koepf, Rajković, and Marinković (2007)]

$$\varepsilon_q^n f(x) = \sum_{k=0}^{n} (-1)^k (1-q)^k \begin{bmatrix} n \\ k \end{bmatrix}_q q^{\binom{k}{2}} x^k D_q^k f(x)$$

$$D_q^n f(x) = \frac{1}{(1-q)^n x^n} \sum_{k=0}^{n} (-1)^k \begin{bmatrix} n \\ k \end{bmatrix}_q q^{\binom{k}{2}-(n-1)k} \varepsilon_q^k f(x).$$

Functions satisfying q-holonomic differential equations or q-holonomic recurrence equations are called *q-holonomic functions*. In particular, q-hypergeometric terms (they satisfy a first-order q-holonomic recurrence equation) as well as the q-exponential functions, q-trigonometric functions [Gasper and Rahman (1990)], [Kac and Cheung (2002)] and the classical q-hypergeometric orthogonal polynomials of the q-Hahn class [Koekoek and Swarttouw (1998)] are all q-holonomic (see Table 1).
The *Maple* package `qFPS` supports all q-functions listed in Table 1, the generalized q-hypergeometric function ${}_r\phi_s$ and all classical q-orthogonal polynomials of the q-Hahn class from [Koekoek and Swarttouw (1998)], and there exist procedures for q-deriving (`qdiff`) and q-shifting (`qshift`) them, where all necessary q-derivative and q-shift rules are implemented.

Example 1. We consider two examples of the procedures `qshift` and `qdiff` from the *Maple* package `qFPS`.

```
> qshift(qsin(x,q),x,q);
```

q-function	definition	q-function	definition
$\exp_q(x)$	${}_1\phi_0\left(\begin{array}{c}0\\ -\end{array}\middle\vert\, q; x\right)$	$\mathrm{e}_q(x)$	${}_1\phi_0\left(\begin{array}{c}0\\ -\end{array}\middle\vert\, q; (1-q)x\right)$
$\mathrm{Exp}_q(x)$	${}_0\phi_0\left(\begin{array}{c}-\\ -\end{array}\middle\vert\, q; -x\right)$	$\mathrm{E}_q(x)$	${}_0\phi_0\left(\begin{array}{c}-\\ -\end{array}\middle\vert\, q; -(1-q)x\right)$
$\sin_q(x)$	$\frac{\exp_q(ix)-\exp_q(-ix)}{2i}$	$\mathrm{s}_q(x)$	$\frac{\mathrm{e}_q(ix)-\mathrm{e}_q(-ix)}{2i}$
$\mathrm{Sin}_q(x)$	$\frac{\mathrm{Exp}_q(ix)-\mathrm{Exp}_q(-ix)}{2i}$	$\mathrm{S}_q(x)$	$\frac{\mathrm{E}_q(ix)-\mathrm{E}_q(-ix)}{2i}$
$\cos_q(x)$	$\frac{\exp_q(ix)+\exp_q(-ix)}{2}$	$\mathrm{c}_q(x)$	$\frac{\mathrm{e}_q(ix)+\mathrm{e}_q(-ix)}{2}$
$\mathrm{Cos}_q(x)$	$\frac{\mathrm{Exp}_q(ix)+\mathrm{Exp}_q(-ix)}{2}$	$\mathrm{C}_q(x)$	$\frac{\mathrm{E}_q(ix)+\mathrm{E}_q(-ix)}{2}$

Table 1. Constitutive q-holonomic functions [Gasper and Rahman (1990)], [Kac and Cheung (2002)]

$$qsin\,(x,q) - x\,qcos\,(x,q)$$

```
> qdiff(qLaguerre(n,alpha,x,q),x,1/q);
```

$$\frac{(q^n-1)\,qLaguerre(n,\alpha,x,q)}{q^{n-1}(q-1)x} - \frac{(q^n q^\alpha - 1)\,qLaguerre(n-1,\alpha,x,q)}{q^{n-1}(q-1)x}$$

3. Determination of q-Holonomic Differential and Recurrence Equations

It is known that sums and products of q-holonomic functions are also q-holonomic and the composition of a q-holonomic function with a power function ax^b is q-holonomic, too. By $\mathcal{H}_q$ we denote the set of q-holonomic functions generated by an iterative application of these procedures starting with the set of *constitutive q-holonomic functions* [1] of Table 1. There exist linear algebra algorithms [Koepf, Rajković, and Marinković (2007)], [Kauers and Koutschan (2009)] based on an ansatz with undetermined coefficients which determine the corresponding q-holonomic differential equation for the sum, the product or the composition directly from the q-holonomic differential equation(s) of the underlying q-holonomic function(s). Those algorithms (which we call the sum, the product and the composition algorithm) and slightly modified versions for q-holonomic recurrence equations, which are more efficient since the product rule for ε_q is much simpler than the one for D_q, are implemented in the *Maple* package `qFPS`. The *Mathematica* package [Kauers and Koutschan (2009)], which deals primarily with q-holonomic sequences, uses related sum and product algorithms.

Example 2. We determine a q-recurrence equation for the function $\cos_q(x)\sin_{1/q}(x)$ via the product algorithm with our *Maple* package `qFPS` [2].

```
> qRE1:=qHolonomicRE(qcos(x,q),F(x));
```

$$qRE1 := q\left(1+x^2\right)F(x) - (1+q)\,Sq_x F(x) + Sq_{x,x} F(x) = 0$$

```
> qRE2:=qHolonomicRE(qsin(x,1/q),F(x));
```

[1] Of course, $\mathcal{H}_q$ can be extended easily by adding new constitutive functions to this list.

[2] Sq_x denotes the q-shift operator ε_q w.r.t. x

$$qRE2 := qF(x) - (1+q)\, Sq_x F(x) + \left(1+q^4x^2\right) Sq_{x,x} F(x) = 0$$

```
> qRE:=qProductRE(qRE1,qRE2,F(x));
```

$$\begin{aligned}qRE := {} & q^6\left(1+x^2\right)\left(1+q^2x^2\right)F(x)\\ & - q^3(1+q)\left(1+q^2\right)\left(1+q^2x^2\right)Sq_xF(x)\\ & + q\left(1+q^2\right)\left(1+q^4x^2\right)\left(1+q+q^2-q^5x^2\right)Sq_{x,x}F(x)\\ & - (1+q)\left(1+q^2\right)\left(1+q^6x^2\right)Sq_{x,x,x}F(x)\\ & + \left(1+q^8x^2\right)\left(1+q^6x^2\right)Sq_{x,x,x,x}F(x) = 0\end{aligned}$$

The resulting q-holonomic recurrence equation `qRE` is a q-holonomic recurrence equation for the product of the functions $\cos_q(x)$ and $\sin_{1/q}(x)$.

However, instead of using these algorithms in the q-FPS algorithm, standard linear algebra techniques are directly applied to the input term in the following Algorithm 1 (`qHolonomicDE`), which is a straightforward q-generalization of the algorithm from [Koepf (1992)], to enforce minimal order of the resulting q-holonomic differential or recurrence equations. This algorithm can be slightly modified to obtain q-holonomic recurrence equations, using the q-shift operator ε_q (`qHolonomicRE`) instead of the Hahn operator D_q. For Algorithm 1 it is important that there exists a q-derivative rule for our q-holonomic function $f(x)$ with the property that the q-derivative can be represented as a linear combination of q-functions from $\mathcal{H}_q$, which are linearly independent over $\mathbb{F}(x)$, preferably of $f(x)$ and its q-shifts itself. This is strongly required in order to put line 6 of the algorithm in a computer algebra system adequately into practice.

Some q-derivatives of special q-functions cannot be easily represented by q-functions from $\mathcal{H}_q$. For those types of functions we have to develop a strategy to express the q-derivatives appropriately and to detect linear dependencies between them. Here, we consider in particular the case of the classical q-orthogonal polynomials $P_n(x)$ of the q-Hahn class. For our implementation we developed q-derivative rules of the following type

$$D_{q^{-1}}P_n(x) = \bar{\alpha}_n(x)P_n(x) + \bar{\beta}_n(x)P_{n-1}(x) \tag{4}$$

with $\bar{\alpha}_n, \bar{\beta}_n \in \mathbb{F}(x, q^n)$, combining the well-known three-term recurrence equation [Koekoek and Swarttouw (1998)]

$$P_{n+1}(x) = (A_nx + B_n)P_n(x) - C_nP_{n-1}(x)$$

with $A_n, B_n, C_n \in \mathbb{F}(q^n)$ and the structure formula [Koepf and Schmersau (2001)]

$$\sigma(x)D_{q^{-1}}P_n(x) = \alpha_nP_{n+1}(x) + \beta_nP_n(x) + \gamma_nP_{n-1}(x)$$

with $\alpha_n, \beta_n, \gamma_n \in \mathbb{F}(q^n)$ and $\sigma(x) \in \mathbb{F}[x]$, which are both valid for each classical q-orthogonal polynomial of the q-Hahn class $P_n(x)$. By expanding $D_{q^{-1}}P_n(x)$ in (4) and solving for $\varepsilon_{q^{-1}}P_n(x)$, we obtain a q-shift rule

$$\varepsilon_{q^{-1}}P_n(x) = \tilde{\alpha}_n(x)P_n(x) + \tilde{\beta}_n(x)P_{n-1}(x) \tag{5}$$

with $\tilde{\alpha}_n, \tilde{\beta}_n \in \mathbb{F}(x, q^n)$. Analogous formulas for D_q or ε_q respectively are gained by substituting (4) or (5) respectively into the well-known (balanced) q-differential equation

$$\bar{r}(x)D_qP_n(x) + \bar{s}(x)P_n(x) + \bar{t}(x)D_{q^{-1}}P_n(x) = 0$$

with $\bar{r}, \bar{s}, \bar{t} \in \mathbb{F}[x]$ or equivalent q-recurrence equation

$$\tilde{r}(x)\varepsilon_qP_n(x) + \tilde{s}(x)P_n(x) + \tilde{t}(x)\varepsilon_{q^{-1}}P_n(x) = 0$$

with $\tilde{r}, \tilde{s}, \tilde{t} \in \mathbb{F}[x]$ respectively and solving for $D_q P_n(x)$ or $\varepsilon_q P_n(x)$ respectively. In order to detect linear dependencies of the polynomials $\ldots, P_{n-1}(x), P_n(x), P_{n+1}(x), \ldots$ in line 6 of Algorithm 1, we use once again the three-term recurrence. Under the above circumstances, the resulting q-holonomic differential equation will have minimal order, if *all* linear dependencies are recognized by the computer algebra system.

Algorithm 1 Determination of a q-holonomic differential equation for a q-holonomic function $0 \not\equiv f(x) \in \mathcal{H}_q$ from its q-derivatives (`qHolonomicDE`)

Input : a q-holonomic function $0 \not\equiv f(x) \in \mathcal{H}_q$
Output : a q-holonomic differential equation for $f(x)$ of minimal order

1 **begin**
2     $n \leftarrow 1$, *done* $\leftarrow$ false
3   **repeat**
4         $DE \leftarrow D_q^n F(x) + \sum_{j=0}^{n-1} A_j(x) D_q^j F(x) = 0$
5         $ansatz \leftarrow D_q^n f(x) + \sum_{j=0}^{n-1} A_j(x) D_q^j f(x)$
6   Decompose *ansatz* in a sum of linearly independent terms over $\mathbb{F}(x)$.
7   Consider the coefficients of linearly independent terms and set them zero.
8   Solve the resulting linear equation system with n unknowns $A_0(x), \ldots, A_{n-1}(x)$ over $\mathbb{F}(x)$.
9   **if** *a nontrivial solution exists* **then**
10   Substitute the solution in DE and multiply with $\operatorname{lcm}(\operatorname{denom}(A_0(x)), \ldots, \operatorname{denom}(A_{n-1}(x)))$.
11   **return** *the q-holonomic differential equation*
12   **else**
13             $n \leftarrow n + 1$
14   **end**
15   **until** *done*
16 **end**

Proof. Throughout the algorithm we deal with q-differential equations which are monic and have rational coefficients $A_j(x) \in F(x)$. In order to get q-holonomic differential equations, we have to multiply the q-differential equation by a common denominator of the coefficients in line 10. The minimality of order is guaranteed by iterating over the order beginning with order 1. Basically, by solving the linear equation system of line 8, we get the (up to a constant factor) uniquely determined q-differential equation, if a nontrivial solution exists. The uniqueness is easy to see. If we consider two monic q-differential equations of the same order, the difference of them is a q-differential equation of smaller order. However, a q-differential equation of smaller order cannot exist, because the algorithm would have detected this q-differential equation in the previous step, so they must be the same. Note, that the **repeat until** loop will eventually end, because the input is a q-holonomic function. Therefore a q-holonomic differential equation must exist, which will be surely found by the algorithm. □

Example 3. Here, we consider the q-functions $f(x) = \sin_q(x)\,\mathrm{Sin}_q(x)$ and $g(x) = \cos_q(x)\,\mathrm{Cos}_q(x)$. With the sum algorithm we obtain a q-holonomic differential equation for $f(x) + g(x)$ of order 3, in contrast to the direct linear algebra approach, which leads to a q-holonomic differential equation of minimal order 1 (because of the identity $f(x) + g(x) = 1$).

```
> qDE1:=qProductDE(qHolonomicDE(qsin(x,q),F(x)),
>                   qHolonomicDE(qSin(x,q),F(x)),F(x)):
> qOrder(qDE1,F(x));
```

$$3$$

```
> qDE2:=qProductDE(qHolonomicDE(qcos(x,q),F(x)),
>                   qHolonomicDE(qCos(x,q),F(x)),F(x)):
> qOrder(qDE2,F(x));
```

$$3$$

```
> qSumDE(qDE1,qDE2,F(x));
```

$$(1+q)\left(q^2+1\right) Dq_x\left(F\left(x\right)\right) + qx\left(q-1\right)\left(1+q\right)\left(q^2+1\right)\left(Dq_{x,x}\right)\left(F\left(x\right)\right) + \left(q^4x^2+1\right)\left(q-1\right)^2\left(Dq_{x,x,x}\right)\left(F\left(x\right)\right) = 0$$

```
> f:=qsin(x,q)*qSin(x,q):
> g:=qcos(x,q)*qCos(x,q):
> qHolonomicDE(f+g,F(x));
```

$$Dq_x\left(F\left(x\right)\right) = 0$$

4. q-Recurrence Equations for q-Series

With the notation of the q-power symbol [Kac and Cheung (2002)]

$$(x \ominus y)_q^j = (x-y)\cdot(x-qy)\cdots(x-q^{j-1}y)$$

we introduce two bases of the linear space of polynomials in x over $\mathbb{F}$, the *q-power basis*

$$\mathfrak{B}^a := \{(x \ominus a)_q^j \mid j \in \mathbb{N}_{\geq 0}\}$$

and the *q-Pochhammer basis*

$$\mathfrak{B}_a := \{(a \ominus x)_q^j \mid j \in \mathbb{N}_{\geq 0}\}.$$

If we consider a q-series w.r.t. one of those bases, we call a the *expansion point* of the q-series. By setting $a=0$, polynomials of the basis $\mathfrak{B}^a$ reduce to regular powers x^j and if we specify $a=1$, polynomials of the basis $\mathfrak{B}_a$ reduce to the q-Pochhammer symbols $(x;q)_j$, explaining their names.

Now, we formulate a *generalized Taylor theorem* ([Kac and Cheung (2002)], [Sprenger (2009)]).

Theorem 1. *A given function $f(x) \in \mathcal{H}_q$ can be expressed as*

$$f(x) = \sum_{j=0}^{\infty} \frac{\left[L^j f(x)\right]_{x=a}}{\prod_{i=1}^{j} \lambda_i} P_j^a(x), \tag{6}$$

if L is a linear operator with $L(1)=0$ and the polynomials $P_j^a(x)$ of degree j satisfy
(a) $P_0^a(a)=1$ and $P_j^a(a)=0$ for all $j \geq 1$
(b) $LP_j^a(x) = \lambda_j P_{j-1}^a(x)$ with $\lambda_j \in \mathbb{F}$ for all $j \geq 1$.

Proof. Applying L j times to both sides of the equation $f(x) = \sum_{k=0}^{\infty} c_k P_k^a(x)$ leads to

$$L^j f(x) = \sum_{k=0}^{\infty} c_k L^j P_k^a(x) \overset{(b)}{=} \sum_{j=1}^{\infty} \lambda_k c_k L^{j-1} P_{k-1}^a(x) \overset{(b)}{=} \ldots \overset{(b)}{=} \sum_{k=j}^{\infty} \lambda_k \cdots \lambda_{k-j+1} c_k P_{k-j}^a(x).$$

Finally, by substituting $x = a$ we get

$$c_j = \frac{\left[L^j f(x)\right]_{x=a}}{\lambda_j \cdots \lambda_{j-j+1} P_0^a(a)} = \frac{\left[L^j f(x)\right]_{x=a}}{\prod_{i=1}^{j} \lambda_i}. \quad \square$$

For $L = \frac{\partial}{\partial x}$ and $\lambda_j = j$ we obtain $P_j^a(x) = (x-a)^j$ and therefore the standard Taylor formula. Now we consider the q-case, which will lead us to the *q-Taylor theorem*. Clearly, elements of $\mathfrak{B}^a$ and $\mathfrak{B}_a$ fulfill property (a). If we assign $L = D_q$ and $\lambda_j = [j]_q$, then the polynomials $P_j^a(x)$ are uniquely determined and we get $P_j^a(x) = (x \ominus a)_q^j$, see also [Jackson (1909)]. On the other hand, if we take the operator $L = D_{q^{-1}}$ and $\lambda_j = -[j]_q$, we get elements of $\mathfrak{B}_a$, once again in a unique way. So these bases arise quite naturally.

4.1. From the Series to the Coefficients

Let $\mathfrak{B} = \{P_j^a(x) \mid j \in \mathbb{N}_{\geq 0}, a \in \mathbb{F}\}$ with
(a) $\deg_x(P_j^a(x)) = j$ for all $j \in \mathbb{N}_{\geq 0}$
(b) $P_j^a(x) \mid P_k^a(x)$ for all $0 \leq j < k$ with $j, k \in \mathbb{N}_{\geq 0}$.
According to [Abramov, Petkovšek, and Ryabenko (2000)], we say that $\mathfrak{B}$ is *compatible with* a given linear operator L, if there exist $i_0, i_1 \in \mathbb{N}_{\geq 0}$ and $a_i \in \mathbb{F}(q^j)$ such that

$$LP_j^a(x) = \sum_{i=-i_0}^{i_1} a_i(q^j)\varepsilon^i P_j^a(x) \text{ with } P_k^a(x) = 0, \text{ if } k < 0, \qquad (7)$$

where ε is the shift operator defined by $\varepsilon j := j + 1$. Table 2 lists our polynomial bases and some of their compatible operators.

L \ $P_j^a(x)$	$(x \ominus a)_q^j$	$(a \ominus x)_q^j$
x	$\left(\varepsilon + aq^j\right)(x \ominus a)_q^j$	$\left(-\frac{1}{q^j}\varepsilon + \frac{a}{q^j}\right)(a \ominus x)_q^j$
D_q	$[j]_q\, \varepsilon^{-1}\,(x \ominus a)_q^j$	-
ε_q	$\left(aq^{j-1}(q^j-1)\varepsilon^{-1} + q^j\right)(x \ominus a)_q^j$	-
$D_{q^{-1}}$	-	$-[j]_q\, \varepsilon^{-1}\,(a \ominus x)_q^j$
$\varepsilon_{q^{-1}}$	-	$\left(\frac{a(q^j-1)}{q^j}\varepsilon^{-1} + \frac{1}{q^j}\right)(a \ominus x)_q^j$

Table 2. Compatible operators L for polynomial bases $\mathfrak{B} = \{P_j^a(x) \mid j \in \mathbb{N}_{\geq 0}, a \in \mathbb{F}\}$

In the following, we assume that $f(x)$ is a q-holonomic function and $\sum_{j=0}^{\infty} c_j P_j^a(x)$ its representation as a q-series w.r.t. a polynomial basis $\mathfrak{B} = \{P_j^a(x) \mid j \in \mathbb{N}_{\geq 0}, a \in \mathbb{F}\}$. We consider only q-holonomic recurrence equations for $f(x)$, because they are easier to handle than q-holonomic differential equations, in particular since the product algorithm is much more efficient because of the linearity of the q-shift operator

$$\varepsilon_q^n(f(x) \cdot g(x)) = \varepsilon_q^n f(x) \cdot \varepsilon_q^n g(x)$$

instead of the complicated q-Leibniz rule for the Hahn operator [Koornwinder (1999)]

$$D_q^n(f(x) \cdot g(x)) = \sum_{k=0}^{n} \begin{bmatrix} n \\ k \end{bmatrix}_q \left(D_q^{n-k} f\right)(q^k x)\left(D_q^k g\right)(x).$$

The second step of the q-FPS algorithm converts a given q-holonomic recurrence equation of $f(x)$ into a q-holonomic recurrence equation for the coefficients c_j, e. g.

$$\sum_{k=0}^{n} \alpha_k(q^j) c_{j+k} = 0 \quad \text{with} \quad \alpha_k \in \mathbb{F}[q^j].$$

For this purpose we define, according to [Abramov, Petkovšek, and Ryabenko (2000)], for a given basis $\mathfrak{B}$ and its compatible operator L the *induced operator of* L *w.r.t.* $\mathfrak{B}$, namely $\mathfrak{R}_{\mathfrak{B}}L$. If (7) holds, then $\mathfrak{R}_{\mathfrak{B}}L := \sum_{i=-i_1}^{i_0} a_{-i}(q^{j+i})\varepsilon^i$. The reason for this definition lies in the fact that L applied to a q-series gives

$$\begin{aligned} L\sum_{j=0}^{\infty} c_j P_j(x) &:= \sum_{j=0}^{\infty} c_j L P_j(x) = \sum_{j=0}^{\infty} \sum_{i=-i_0}^{i_1} a_i(q^j) c_j \varepsilon^i P_j(x) \\ &= \sum_{j=0}^{\infty} \sum_{i=-i_0}^{i_1} a_i(q^{j-i}) c_{j-i} P_j(x) = \sum_{j=0}^{\infty} \sum_{i=-i_1}^{i_0} a_{-i}(q^{j+i}) c_{j+i} P_j(x). \end{aligned}$$

In [Abramov, Petkovšek, and Ryabenko (2000)] it is shown that the set of all compatible operators for a given basis $\mathfrak{B}$ is an $\mathbb{F}$-algebra, i.e. there exists a simple pattern-matching algorithm for every q-holonomic equation involving an operator L which is compatible to $\mathfrak{B}$, which efficiently determines the induced equation. This leads to the following algorithm for computing $\mathfrak{R}_{\mathfrak{B}}L$ or the corresponding recurrence equation $(\mathfrak{R}_{\mathfrak{B}}L)(c_j) = 0$ respectively.

Algorithm 2 Determination of a q-holonomic recurrence equation for the coefficients of a q-series (w.r.t. $\mathfrak{B}^a$ or $\mathfrak{B}_a$ resp.) from a q-holonomic recurrence equation of a q-series (`qREtoRE` with `base=qpower` or `base=qpochhammer` resp. and `expansionpt=a`)

Input : a q-holonomic recurrence equation $\sum_{k=0}^{n} a_k(x)\varepsilon_q^k f(x) = 0$, a basis $\mathfrak{B}$ and an expansion point a

Output : a q-holonomic recurrence equation for c_j, where $f(x) = \sum_{j=0}^{\infty} c_j P_j^a(x)$

begin
- $RE \leftarrow \sum_{k=0}^{n} a_k(x)\varepsilon_q^k f(x) = 0$
- $RE \leftarrow$ expand the left hand side of RE
- Determine $\beta_{ki}(c_j) = \beta_{ki}^{(k+i)}(c_j)$ for every $\alpha_{ki}x^i\varepsilon_q^k f(x)$ from RE by induction w.r.t. $\mathfrak{B}$ with formulas (9), which follow from the corresponding entries of Table 2
- $RE \leftarrow$ Substitute $x^i\varepsilon_q^k f(x)$ in RE by $\beta_{ki}(c_j)$ and multiply with the least common multiple of the denominator
- $RE \leftarrow$ Shift RE such that only positive shifts occur
- **return** RE

end

Proof. Let

$$Lf(x) := \sum_{k=0}^{n} a_k(x)\varepsilon_q^k f(x) = \sum_{k=0}^{n} \sum_{i=0}^{m_k} \alpha_{ki} x^i \varepsilon_q^k f(x) = 0 \tag{8}$$

be the given q-holonomic recurrence equation. Because of compatibility the equations

$$\varepsilon_q P_j^a(x) = \sum_{s=-i_0}^{i_1} \tilde{a}_s(q^j)\varepsilon^s P_j^a(x)$$

and

$$x\,P_j^a(x) = \sum_{t=-\tilde{i}_0}^{\tilde{i}_1} \tilde{b}_t(q^j)\varepsilon^t P_j^a(x)$$

with $\tilde{a}_s, \tilde{b}_t \in \mathbb{F}(q^j)$ are valid. Then after multiplying with c_j and summing we obtain

$$\varepsilon_q f(x) = \sum_{j=0}^{\infty} \sum_{s=-i_1}^{i_0} \tilde{a}_{-s}(q^{j+s}) c_{j+s} P_j^a(x)$$

and

$$x\,f(x) = \sum_{j=0}^{\infty} \sum_{t=-\tilde{i}_1}^{\tilde{i}_0} \tilde{b}_{-t}(q^{j+t}) c_{j+t} P_j^a(x).$$

We define the following recurrence

$$\begin{aligned} \beta_{ki}^{(l)}(c_j) &:= \sum_{s=-i_1}^{i_0} \tilde{a}_{-s}(q^{j+s})\varepsilon^s \beta_{ki}^{(l-1)}(c_j) \text{ for } l = 1, \ldots, k, \\ \beta_{ki}^{(l)}(c_j) &:= \sum_{t=-\tilde{i}_1}^{\tilde{i}_0} \tilde{b}_{-t}(q^{j+t})\varepsilon^t \beta_{ki}^{(l-1)}(c_j) \;\text{ for } l = k+1, \ldots, k+i \end{aligned} \tag{9}$$

with initial value $\beta_{ki}^{(0)}(c_j) := c_j$. Obviously, (8) is equivalent to

$$\sum_{j=0}^{\infty} \underbrace{\sum_{k=0}^{n} \sum_{i=0}^{m_k} \alpha_{ki}\beta_{ki}(c_j)}_{\Rightarrow (\mathfrak{R}_{\mathfrak{B}} L)(c_j)=0} P_j^a(x) = 0$$

with

$$\beta_{ki}(c_j) := \beta_{ki}^{(k+i)}(c_j) = \sum_{r=-I_0}^{I_1} p_{kir}(q^j)\varepsilon^r c_j,$$

$p_{kir} \in \mathbb{F}(q^j)$ and $I_0, I_1 \in \mathbb{N}_{\geq 0}$. $\square$

Example 4. We continue with Example 2 and determine a recurrence equation for the coefficients c_j of the q-series of $\cos_q(x)\sin_{1/q}(x)$ w.r.t. to $\mathfrak{B}^0$, i.e. $\cos_q(x)\sin_{1/q}(x) = \sum_{j=0}^{\infty} c_j x^j$.

```
> RE:=qREtoRE(qRE,F(x),c(j),base=qpower,expansionpt=0);
```

$$RE := q^2\left(1+qq^j\right)\left(1+q^2q^j\right)c(j) + \left(1+q^2\right)\left(1+q^5\left(q^j\right)^2\right)c(j+2) + \left(1-q^4q^j\right)\left(1-q^3q^j\right)c(j+4) = 0$$

4.2. From the Coefficients to the Series

Analogously to (7), we can define compatibility as

$$LP_j(x) = \sum_{i=-i_0}^{i_1} a_i(x)\varepsilon_q^i P_j(x) \text{ with } P_k(x) = 0, \text{ if } k < 0$$

with $i_0, i_1 \in \mathbb{N}_{\geq 0}$ and $a_i \in \mathbb{F}(x)$. Then we use the identities of Table 3 and get an inverse algorithm of Algorithm 2, which uses a similar recurrence as (9).

L \ $P_j^a(x)$	$(x \ominus a)_q^j$	$(a \ominus x)_q^j$
q^j	$\frac{1}{qx-a}\varepsilon_q \boldsymbol{\varepsilon}\,(x \ominus a)_q^j$	$\left(\frac{x-a}{x}\varepsilon_q + \frac{a}{x}\right)(a \ominus x)_q^j$
ε^{-1}	$\left(\frac{a}{(qx-a)x}\varepsilon_q + \frac{1}{x}\right)(x \ominus a)_q^j$	$\frac{q}{qa-x}\varepsilon_q^{-1}\,(a \ominus x)_q^j$

Table 3. Compatible operators L for polynomial bases $\mathfrak{B} = \{P_j^a(x) \mid j \in \mathbb{N}_{\geq 0}, a \in \mathbb{F}\}$

In case of the q-power basis, however, there is no pure compatibility, because of the occurring shift operator $\boldsymbol{\varepsilon}$ in Table 3, so the algorithm needs to be slightly adapted. In this case, for $i, k \in \mathbb{N}_0$, we have

$$\begin{aligned}
\sum_{j=0}^{\infty} \left(q^j\right)^i (\varepsilon^k c_j)\,(x \ominus a)_q^j &= \sum_{j=-1}^{\infty} \left(q^j\right)^{i-1} (\varepsilon^k c_j)\, q^j\, (x \ominus a)_q^j \\
&= \sum_{j=-1}^{\infty} \left(q^j\right)^{i-1} (\varepsilon^k c_j)\frac{1}{qx-a}\varepsilon_q \varepsilon\,(x \ominus a)_q^j \\
&= \frac{q^{-(i-1)}}{qx-a}\varepsilon_q \sum_{j=0}^{\infty} \left(q^j\right)^{i-1} (\varepsilon^{k-1} c_j)\,(x \ominus a)_q^j \\
&= \ldots \text{(iterating w.r.t. } i) \\
&= (-1)^i \frac{q^{-\binom{i}{2}}}{(a \ominus qx)_q^i}\varepsilon_q^i \sum_{j=0}^{\infty} (\varepsilon^{k-i} c_j)\,(x \ominus a)_q^j\,. \qquad (10)
\end{aligned}$$

Roughly speaking, we first eliminate all powers $\left(q^j\right)^i$ in the corresponding operator of the given recurrence equation

$$\sum_{k=0}^{n} \sum_{i=0}^{m_k} \alpha_{ki} (q^j)^i \varepsilon^k c_j = 0 \qquad (\alpha_{ki} \in \mathbb{F}) \qquad (11)$$

with the help of equation (10). In order to get only nonnegative shifts we require that $k - i \in \mathbb{N}_0$ for $i, k \in \mathbb{N}_0$. This can be achieved by shifting the expanded recurrence equation (11) by $j \to j - \min\{k - m_k \in -\mathbb{N}_0 \mid k = 0, \ldots, n\}$. The positive shifts, which are left, will be eliminated with the corresponding entry of Table 3, similar to Algorithm 2 (for details see [Sprenger (2009)]). With these modifications we can now determine a q-holonomic recurrence equation for the series, given a q-holonomic recurrence for the coefficients c_j. In fact, for $a = 0$ this algorithm is the inverse function of Algorithm 2.

We apply this algorithm to the generalized q-hypergeometric function ${}_r\phi_s$ and are now able to determine a q-holonomic recurrence equation for those ${}_r\phi_s$, which can be written as q-series w.r.t. $\mathfrak{B}^a$ or $\mathfrak{B}_a$ in a simple way. First of all, we check if the given ${}_r\phi_s$ can be represented as q-series w.r.t. one of the bases or not by analyzing the upper and lower parameters and the argument of ${}_r\phi_s$. If it can be represented as q-series w.r.t. $\mathfrak{B}^a$ or $\mathfrak{B}_a$, we determine the underlying basis and the expansion point. Secondly, we can get the corresponding q-holonomic recurrence of first order for the coefficients from the term ratio ${c_{j+1}}/{c_j}$ and then apply Algorithm 2 properly to our recurrence equation. In detail, this yields the following simple algorithm, which only uses rewriting techniques for the determination of the q-holonomic recurrence equation. If the algorithm fails, there could be nevertheless a q-holonomic recurrence equation for ${}_r\phi_s$ (but only for q-series which are no

q-series w.r.t. $\mathfrak{B}^a$ or $\mathfrak{B}_a$). In order to deduce a q-holonomic recurrence for the generalized q-hypergeometric function in general, the q-Zeilberger algorithm ([Koornwinder (1993)], [Böing and Koepf (1999)]) should be preferred, which is much more powerful.

Algorithm 3 Determination of a q-holonomic recurrence equation for the generalized q-hypergeometric function ${}_r\phi_s$ (`qHolonomicRE` for `qphihypergeom`)

Input : a generalized q-hypergeometric function ${}_r\phi_s(\mathbf{a}, \mathbf{b} \mid q; X)$
Output : a q-holonomic recurrence equation for ${}_r\phi_s$

begin
  switch ${}_r\phi_s(\mathbf{a}, \mathbf{b} \mid q; X)$ **do**
    case *x is in numerator of argument X and none of upper parameters $\mathbf{a}$ contains x*
      $c_j \leftarrow \frac{(\mathbf{a};q)_j}{(\mathbf{b};q)_j} \frac{X^j}{(q;q)_j} \left((-1)^j q^{\binom{j}{2}}\right)^{1+s-r} \frac{1}{x^j}$, *basis* $\leftarrow \mathfrak{B}^0$
    end
    case *x is in numerator of argument X and denominator of one upper parameter a contains x*
      $c_j \leftarrow \frac{(\mathbf{a};q)_j}{(\mathbf{b};q)_j} \frac{X^j}{(q;q)_j} \left((-1)^j q^{\binom{j}{2}}\right)^{1+s-r} \frac{1}{x^j} (a;q)_j^{-1}$, *basis* $\leftarrow \mathfrak{B}^{ax}$
    end
    case *x is not in argument X and numerator of one upper parameter a contains x*
      $c_j \leftarrow \frac{(\mathbf{a};q)_j}{(\mathbf{b};q)_j} \frac{X^j}{(q;q)_j} \left((-1)^j q^{\binom{j}{2}}\right)^{1+s-r} \left(\frac{a}{x}\right)^j (a;q)_j^{-1}$, *basis* $\leftarrow \mathfrak{B}_{\frac{x}{a}}$
    end
    otherwise
      return *"No q-holonomic recurrence equation could be found."*
    end
  end
  $RE \leftarrow$ Construct the q-holonomic recurrence equation for c_j of order 1 out of the term ratio ${c_{j+1}}/{c_j}$
  $RE \leftarrow$ Convert RE in a q-holonomic recurrence for the q-series w.r.t. *basis* with the inverse algorithm of Algorithm 2
  return RE
end

Proof. The equations

$$x^j \left(\frac{a}{x}; q\right)_j = (x \ominus a)_q^j \quad \text{and} \quad (ax;q)_j = a^j \left(\frac{1}{a} \ominus x\right)_q^j$$

were applied (the latter for $a \neq 0$), in order to detect which basis to use in the representation of ${}_r\phi_s$ as a q-series. □

With Algorithm 3, we can determine particularly q-holonomic recurrence equations for all classical q-orthogonal polynomials of the q-Hahn class directly from their representation as generalized q-hypergeometric function. Any classical q-orthogonal polynomial can be represented as a q-hypergeometric series w.r.t. at least one of the bases $\mathfrak{B}^a$ or $\mathfrak{B}_a$ and therefore Algorithm 3 leads definitely to the desired recurrence equation.

For the generalized q-hypergeometric function, which is a q-series w.r.t. the trivial basis $\mathfrak{B}^0 = \{x^j \mid j \in \mathbb{N}_{\geq 0}\}$, we can state its recurrence equation explicitly as

$$\Bigg((-1)^{s+1} q^{r-m} x \varepsilon_q^{1+s-m} \left(\prod_{i=1}^{r} (1 - a_i \varepsilon_q)\right) + (-1)^r (\varepsilon_q - 1) \varepsilon_q^{r-m} \left(\prod_{i=1}^{s} (1 - q^{-1} b_i \varepsilon_q)\right)\Bigg) f(x) = 0,$$

where $m = \min(r, s+1)$. Obviously, the order of this recurrence is $\max(r, s+1)$. One can prove the q-holonomic recurrence equation directly out of the recurrence equation for the coefficients c_j of the generalized q-hypergeometric function

$$(1 - q^j) \prod_{i=1}^{s} (1 - b_i q^j) c_{j+1} - (-q^k)^{1+s-r} \prod_{i=1}^{r} (1 - a_i q^j) x c_j = 0$$

by multiplying x^j, applying the rules of Table 3 and summing w.r.t. j, similar to the proof of Theorem 2.1 in [Koepf (1992)].

5. q-Hypergeometric Solutions of q-Recurrence Equations

The third step of the q-FPS algorithm is to determine all q-hypergeometric solutions of the q-holonomic recurrence equation. For this purpose we use the most efficient algorithm by Horn [Horn (2008)] which is based on the q-Petkovšek algorithm [Abramov, Paule, and Petkovšek (1998)]. In the difference case by far the fastest algorithm known is van Hoeij's algorithm [van Hoeij (1998)], [Cluzeau and van Hoeij (2005)]. However, a q-analogue [Horn (2008)] of this version turns out to be (in most cases) not as efficient as Horn's variant of q-Petkovšek's algorithm. In this modified version of the q-Petkovšek algorithm the *q-Newton polygon* and its characteristic equations are used to reduce the numbers of possible solutions significantly.

Example 5. Continuing with Example 4, we determine all q-hypergeometric solutions of the recurrence equation of fourth order with the modified q-Petkovšek algorithm and obtain the linear combination [3]

```
> qHypergeomSolveRE(RE,c(j));
```

$$A_1 i^j + (-1)^j A_2 i^j + \frac{1}{2} A_3 \frac{i^j \left(1 + q^j\right) \mathit{qpochhammer}\,(-1, q, j)}{\mathit{qpochhammer}\,(q, q, j)} + (-1)^j \frac{1}{2} A_4 \frac{i^j \left(1 + q^j\right) \mathit{qpochhammer}\,(-1, q, j)}{\mathit{qpochhammer}\,(q, q, j)}$$

This part, together with the second part of the q-FPS algorithm which we described in the previous section, can be regarded as a kind of an inverse q-Zeilberger algorithm [Koornwinder (1993)], [Böing and Koepf (1999)]. Given a q-hypergeometric term $F(n, k)$ in n and k with finite support (i.e. $F(\tilde{n}, \tilde{k}) \neq 0$ only for finitely many $(\tilde{n}, \tilde{k}) \in \mathbb{Z}^2$), the q-Zeilberger algorithm determines a q-holonomic recurrence equation for $S_n := \sum_k F(n, k)$.

[3] one can use the classical q-Petkovšek algorithm by specifying the option `method=qPetkovsek` and the q-van Hoeij algorithm by `method=qVanHoeij` to `qHypergeomSolveRE`. The default option is `method=modqPetkovsek`.

If S_n is given by its recurrence equation or as a q-holonomic function and it can be represented as q-hypergeometric series w.r.t. one of our bases, the q-FPS algorithm surely calculates a sum representation and therefore a summand $F(n,k)$. The initial values can be computed by the method described in the forthcoming section.

6. The *q*-FPS Algorithm

We sum up the three algorithms of the above sections and give an overview of the basic q-FPS algorithm.

Algorithm 4 Determination of a (linear combination of) q-hypergeometric series (w.r.t. $\mathfrak{B}^a$ or $\mathfrak{B}_a$ resp.) for a q-holonomic function (`` `convert/qFPS` `` with `base=qpower` or `base=qpochhammer` resp. and `expansionpt=a`)

Input : a q-holonomic function $f(x) \in \mathcal{H}_q$, a basis $\mathfrak{B}$ and an expansion point a
Output : a (linear combination of) q-hypergeometric series $\sum_{j=0}^{\infty} c_j P_j^a(x)$ for $f(x)$

1 **begin**
2 $qRE \leftarrow$ Determine a q-holonomic recurrence equation for $f(x)$ with Algorithm 1 (for recurrence equations)
3 $RE \leftarrow$ Convert qRE in a q-holonomic recurrence equation for c_j w.r.t. $\mathfrak{B}$ and expansion point a with Algorithm 2
4 Determine all m-fold q-hypergeometric solutions of RE (if RE is of q-hypergeometric type) or (if this fails) all q-hypergeometric solutions of RE with the modified q-Petkovšek algorithm
5 Determine with sufficiently many initial values (out of the q-Taylor theorem) the q-series with the above solutions
6 **return** *the resulting (linear combination of) q-hypergeometric series*
7 **end**

In the fourth line we first try to find m-fold q-hypergeometric solutions, if the recurrence is of q-hypergeometric type, i.e. the recurrence equation can be written as

$$a_m(q^j)c_{j+m} + a_0(q^j)c_j = 0 \quad \text{or} \quad \frac{c_{j+m}}{c_j} = -\frac{a_0(q^j)}{a_m(q^j)}.$$

From the q-Taylor theorem we know that

$$c_j = \frac{\left[L^j f(x)\right]_{x=a}}{\prod_{i=1}^{j} \lambda_i} \quad \text{with} \quad \lambda_i = \frac{LP_i^a(x)}{P_{i-1}^a(x)}. \tag{12}$$

Inductively we determine the m solutions

$$c_{mj}, c_{mj+1}, \ldots, c_{mj+m-1}$$

by using sufficiently many initial values, which we get from (12). Thus, we obtain the following representation for $f(x)$

$$f(x) = \sum_{j=0}^{\infty} c_{mj} P_{mj}^a(x) + \sum_{j=0}^{\infty} c_{mj+1} P_{mj+1}^a(x) + \cdots + \sum_{j=0}^{\infty} c_{mj+m-1} P_{mj+m-1}^a(x).$$

If there are no m-fold q-hypergeometric solutions, we try to find q-hypergeometric solutions via the modified q-Petkovšek algorithm. Let

$$\left\{c_j^{(1)}, c_j^{(2)}, \ldots, c_j^{(l)}\right\}$$

be a basis of q-hypergeometric solutions of the q-holonomic recurrence for the coefficients c_j with order n and $l \leq n$. Then we set up a linear equation system in the unknowns $t_k \in \mathbb{F}$ with the help of n initial values (from the q-Taylor theorem)

$$\sum_{k=1}^{l} t_k c_r^{(k)} = \frac{\left[L^r f(x)\right]_{x=a}}{\prod_{i=1}^{r} \lambda_i} \quad \text{with} \quad r = 0, \ldots, n-1$$

and we try to solve it. If a solution exists, we get the q-series expansion

$$f(x) = \sum_{j=0}^{\infty} \left(\sum_{k=1}^{l} t_k c_j^{(k)} \right) P_j^a(x) = \sum_{k=1}^{l} t_k \left(\sum_{j=0}^{\infty} c_j^{(k)} P_j^a(x) \right),$$

which in general is not q-hypergeometric, but a linear combination of q-hypergeometric series. This method requires that the initial values are defined for $r = 0, \ldots, n-1$ and that there exists at least one nonvanishing initial value. We achieve this by the following strategy. We directly determine an index i, such that c_j is defined for all $j \geq i$ from the recurrence equation. Let

$$Lc_j := \sum_{k=0}^{n} a_k(q^j) \varepsilon^k c_j = 0$$

be given with $a_k \in \mathbb{F}[q^j]$ and c_j a q-hypergeometric solution. Furthermore, let

$$r(q^j) := \frac{c_{j+1}}{c_j} = \frac{s(q^j)}{t(q^j)} \quad \text{with} \quad s, t \in \mathbb{F}[q^j]$$

be the q-certificate of c_j with $\gcd(s(q^j), t(q^j)) = 1$. This is equivalent to the fact, that the operator L has a right factor $t(q^j)\varepsilon - s(q^j)$. Let $\tilde{L} := \sum_{k=0}^{n-1} b_k(q^j)\varepsilon^k$ be the appropriate operator, such that $L = \tilde{L} \cdot (t(q^j)\varepsilon - s(q^j))$ is valid. By expanding we get

$$L = \underbrace{b_{n-1}(q^j)\varepsilon^{n-1}(t(q^j))}_{a_n(q^j)} \varepsilon^n + \cdots + \underbrace{\left(-b_0(q^j)s(q^j)\right)}_{a_0(q^j)}.$$

Hence, it follows that

$$s(q^j) \mid a_0(q^j) \qquad \text{and} \qquad t(q^j) \mid a_n(q^{j-n+1}).$$

Let $M := \{k \in \mathbb{N}_{\geq 0} \mid q^k \text{ is a zero of } a_0(q^j)a_n(q^{j-n+1})\}$. Obviously, M contains all $k \in \mathbb{N}_{\geq 0}$, for which $r(q^j)$ has zeros and poles of the form q^k. Therefore we have $i = 1 + \max\left(M \cup \{-1\}\right)$. For this reason, we consider $\tilde{r}(q^j) := r(q^{j+i})$ and get a q-certificate of a q-hypergeometric term $\tilde{c}_j$, which is defined for all $j \geq 0$.

Example 6. First, we determine the corresponding q-series for the function $\cos_q(x)\sin_{1/q}(x)$ used in the previous examples. Then we show, how one can easily determine further q-series representations of q-holonomic functions by a single function call using `qFPS`. If no basis and no expansion point is specified in `` `convert/qFPS` `` (implementation of the q-FPS algorithm in `qFPS`), then the options `base=qpower` and `expansionpt=0` are chosen[4].

```
> convert(qcos(x,q)*qsin(x,1/q),qFPS);
```

$$\sum_{k=0}^{\infty} \frac{1}{4} \frac{(-1)^k \left(2\,(q;q)_{2k+1} - \left(1+q^{2\,k+1}\right)(-1;q)_{2k+1}\right) x^{2\,k+1}}{(q;q)_{2k+1}}$$

By simplification we obtain the following q-series expansion of $\cos_q(x)\sin_{1/q}(x)$ w.r.t. $\mathfrak{B}^0$

$$\frac{1}{2}\sum_{k=0}^{\infty}(-1)^k \frac{(q;q)_{2k+1} - (-q;q)_{2k+1}}{(q;q)_{2k+1}} x^{2k+1}.$$

Sums with finite support ($n \in \mathbb{N}_0$)

```
> PS:=convert(x^n,qFPS,x,expansionpt=1);
```

$$PS := \sum_{k=0}^{\infty} \frac{(-1)^k q^{1/2\,k(2\,n-k+1)} \left(q^{-n};q\right)_k (x \ominus 1)_q^k}{(q;q)_k}$$

```
> convert(PS,qbinomial);
```

$$\sum_{k=0}^{\infty} \mathit{qbinomial}\,(n,k,q)\,(x \ominus 1)_q^k$$

```
> convert(qLaguerre(n,alpha,x,q),qFPS,x,base=qpochhammer,expansionpt=-1);
```

$$\sum_{k=0}^{\infty} \frac{(-1)^k q^{k(n+\alpha+1)} \left(q^{-n};q\right)_k (-1 \ominus x)_q^k}{(q;q)_n\,(q;q)_k}$$

q-Hypergeometric series

```
> convert(qexp(x,q),qFPS,base=qpochhammer,expansionpt=a,var=1/q);
```

$$\sum_{k=0}^{\infty} \frac{\mathit{qexp}\,(a,q)\,q^{-k}\,(a \ominus x)_{q^{-1}}^k}{\left(q^{-1};q^{-1}\right)_k}$$

```
> convert(qsin(x,q)+qSin(x,1/q),qFPS);
```

$$\sum_{k=0}^{\infty} -\frac{(-1)^k x^{2\,k+1}\left(q^{2\,k+1}-1\right)}{(q;q)_{2\,k+1}}$$

Linear combinations of q-hypergeometric series

```
> convert(qsin(x,q),qFPS,expansionpt=a);
```

$$\sum_{k=0}^{\infty} \frac{(-1)^k\,\mathit{qsin}\,(a,q)\,(x \ominus a)_q^{2k}}{(q;q)_{2k}} + \sum_{k=0}^{\infty} \frac{(-1)^k\,\mathit{qcos}\,(a,q)\,(x \ominus a)_q^{2k+1}}{(q;q)_{2k+1}}$$

```
> convert(sinq(x,q)*qExp(x,q),qFPS);
```

$$\sum_{k=0}^{\infty} \frac{1}{2} i \frac{\mathrm{e}^{1/2ik\pi}}{\mathit{qfactorial}\,(k,q)} \left((-1)^k \left(\frac{i}{q-1};q\right)_k - \left(\frac{-i}{q-1};q\right)_k\right) x^k$$

```
> convert((a*x+b)*qcos(x,q)+(c*x+d)*qsin(x,q),qFPS,x);
```

$$\sum_{k=0}^{\infty} \frac{(-1)^k \left(d+a-aq^{2\,k+1}\right) x^{2\,k+1}}{(q;q)_{2k+1}} + \sum_{k=0}^{\infty} \frac{(-1)^k \left(b-c+cq^{2\,k}\right) x^{2\,k}}{(q;q)_{2k}}$$

[4] For the sake of readability, we use the short notation for q-Pochhammer symbols $(a;q)_k$ and q-powers $(a \ominus b)_q^k$ instead of the long *Maple* versions `qpochhammer(a,q,k)` and `qpower(a,b,q,k)`.

7. The *Maple* Package `qFPS`

The main procedures of `qFPS` are `qshift`, `qdiff`, `qHolonomicRE`, `qHolonomicDE`, `qHypergeomSolveRE`, the conversion procedures `qREtoRE` and `REtoqRE`, and the most important procedure `` `convert/qFPS` ``, which we have seen in examples before. The procedures for the q-holonomic algebra are `qSumRE`, `qProductRE` and `qCompositionRE`. For all those procedures corresponding `DE`-variants are available. The existing q-functions are listed in Table 4. For every q-function q-shift and q-derivative rules and function values at specific points are available. Additionally, for every family of classical q-orthogonal polynomials a three-term recurrence equation is stored for the usage in `qHolonomicRE`. The generalized q-hypergeometric function

$$ {}_r\phi_s\left(\begin{array}{c} a_1,\ldots,a_r \\ b_1,\ldots,b_s \end{array} \middle| \, q;x\right) $$

can be invoked by the command `qphihypergeom([a1,...,ar],[b1,...,bs],x,q)`.

One can extend this family of constituting functions via the procedures `addqshift`, `addqiff`, `addqrec` (see the accompanying demo worksheet).

8. Applications of the q-FPS Algorithm

In this section we present several algorithmic proofs. The main idea of these types of proofs go back to [Zeilberger (1990)]. With the q-FPS algorithm, we are now able to deduce or verify easily and automatically special function identities like summation formulas [Gasper and Rahman (1990)], [Koekoek and Swarttouw (1998)], e.g.

$$ \frac{\left(\frac{c}{b};q\right)_n}{(c;q)_n} b^n = {}_2\phi_1\left(\begin{array}{c} q^{-n},b \\ c \end{array} \middle| \, q;q\right), \tag{13} $$

transformation formulas [Gasper and Rahman (1990)], [Koekoek and Swarttouw (1998)], e.g. Jackson's formula

$$ \frac{\left(\frac{q}{b};q\right)_\infty}{\left(\frac{q}{b}q^{-n};q\right)_\infty} {}_2\phi_1\left(\begin{array}{c} q^{-n},\frac{c}{a} \\ c \end{array} \middle| \, q;\frac{aq}{b}\right) = {}_3\phi_2\left(\begin{array}{c} q^{-n},a,0 \\ b,c \end{array} \middle| \, q;q\right), $$

q-hypergeometric series representations of Taylor coefficients of generating functions [Koepf (2003)], e.g.

$$ \frac{1}{(t;q)_\infty} {}_1\phi_1\left(\begin{array}{c} -x \\ 0 \end{array} \middle| \, q;q^{\alpha+1}t\right) = \sum_{n=0}^{\infty} \mathrm{L}_n^{(\alpha)}(x;q)\, t^n, $$

simple identities like

q-function	*Maple* call
$(x \ominus a)_q^k$	`qpower(x,a,q,k)`
$(x \ominus a)_q^\infty$	`qpower(x,a,q,infinity)`
$(x;q)_k$	`qpochhammer(x,q,k)`
$(x;q)_\infty$	`qpochhammer(x,q,infinity)`
$\exp_q(x)$ / $\mathrm{e}_q(x)$	`qexp(x,q) / expq(x,q)`
$\mathrm{Exp}_q(x)$ / $\mathrm{E}_q(x)$	`qExp(x,q) / Expq(x,q)`
$\sin_q(x)$ / $\mathrm{s}_q(x)$	`qsin(x,q) / sinq(x,q)`
$\mathrm{Sin}_q(x)$ / $\mathrm{S}_q(x)$	`qSin(x,q) / Sinq(x,q)`
$\cos_q(x)$ / $\mathrm{c}_q(x)$	`qcos(x,q) / cosq(x,q)`
$\mathrm{Cos}_q(x)$ / $\mathrm{C}_q(x)$	`qCos(x,q) / Cosq(x,q)`
$\mathrm{U}_n^{(a)}(x;q)$	`AlSalamCarlitzI(n,a,x,q)`
$\mathrm{V}_n^{(a)}(x;q)$	`AlSalamCarlitzII(n,a,x,q)`
$\mathrm{C}_n\,(x;a;q)$	`qCharlier(n,a,x,q)`
$\mathrm{K}_n\,(x;a;q)$	`AlternativeqCharlier(n,a,x,q)`
$\mathrm{h}_n\,(x;q)$	`DiscreteqHermiteI(n,x,q)`
$\tilde{\mathrm{h}}_n\,(x;q)$	`DiscreteqHermiteII(n,x,q)`
$\mathrm{p}_n\,(x;a,b\|q)$	`LittleqJacobi(n,a,b,x,q)`
$\mathrm{P}_n\,(x;a,b,c;q)$	`BigqJacobi(n,a,b,c,x,q)`
$\mathrm{p}_n\,(x\|q)$	`LittleqLegendre(n,x,q)`
$\mathrm{P}_n\,(x;c;q)$	`BigqLegendre(n,c,x,q)`
$\mathrm{p}_n\,(x;a\|q)$	`LittleqLaguerre(n,a,x,q)`
$\mathrm{P}_n\,(x;a,b;q)$	`BigqLaguerre(n,a,b,x,q)`
$\mathrm{L}_n^{(\alpha)}(x;q)$	`qLaguerre(n,alpha,x,q)`
$\mathrm{S}_n\,(x;q)$	`StieltjesWigert(n,x,q)`
$\mathrm{Q}_n\,(x;a,b;N\|q)$	`qHahn(n,a,b,N,x,q)`
$\mathrm{M}_n\,(x;b,c;q)$	`qMeixner(n,b,c,x,q)`
$\mathrm{K}_n\,(x;p,N;q)$	`qKrawtchouk(n,p,N,x,q)`
$\mathrm{K}_n^{\mathrm{aff}}\,(x;p,N;q)$	`AffineqKrawtchouk(n,p,N,x,q)`
$\mathrm{K}_n^{\mathrm{qtm}}\,(x;p,N;q)$	`QuantumKrawtchouk(n,p,N,x,q)`

Table 4. q-Functions and their names and arguments in `qFPS`

$$\begin{aligned}
\exp_q(x) &= \mathrm{e}_q\left(\frac{x}{1-q}\right) = \frac{1}{(x;q)_\infty} = {}_1\phi_0\left(\begin{matrix}0\\-\end{matrix}\,\middle|\, q;x\right) = \sum_{j=0}^\infty \frac{x^k}{(q;q)_k} \\
&= \exp_q(a)\,{}_1\phi_0\left(\begin{matrix}\frac{a}{x}\\-\end{matrix}\,\middle|\, q;x\right) = \exp_q(a)\,{}_1\phi_1\left(\begin{matrix}\frac{a}{x}\\\frac{a}{q}\end{matrix}\,\middle|\, \frac{1}{q};\frac{x}{q}\right) \\
&= \exp_q(a)\,{}_2\phi_1\left(\begin{matrix}\frac{x}{a},0\\\frac{q}{a}\end{matrix}\,\middle|\, q;q\right) = \exp_q(a)\,{}_1\phi_0\left(\begin{matrix}\frac{x}{a}\\-\end{matrix}\,\middle|\, \frac{1}{q};\frac{a}{q}\right)
\end{aligned}$$

or the q-addition theorems

$$\sin_q(x)\,\mathrm{Sin}_q(x) + \cos_q(x)\,\mathrm{Cos}_q(x) = 1$$

and

$$\sin_q(x)\,\mathrm{Cos}_q(x) - \mathrm{Sin}_q(x)\cos_q(x) = 0.$$

All these deductions are easily done from left to right with `qFPS` and presented in detail in [Sprenger (2009)] or in the *Maple* worksheet accompanying this paper (to be downloaded from `http://www.mathematik.uni-kassel.de/~koepf/Publikationen`).

In the following example we use a `qFPS` procedure called `` `convert/qphihypergeom` ``, which converts a q-hypergeometric series $\sum_j c_j P_j^a(x)$ in its representation as generalized q-hypergeometric function. This is done (roughly) by factorizing numerator and denominator of ${c_{j+1}P_{j+1}^a(x)}/{c_j P_j^a(x)}$ and reading off parameters (every linear factor representing one q-Pochhammer symbol).

Example 7. We deduce the q-Chu-Vandermonde identity (13) exemplarily.

```
> term:=qpochhammer(c/b,q,n)/qpochhammer(c,q,n)*b^n;
```

$$term := \frac{\left(\frac{c}{b};q\right)_n}{(c;q)_n} b^n$$

```
> PS:=convert(term,qFPS,b,base=qpochhammer,expansionpt=1);
```

$$PS := \sum_{k=0}^{\infty} \frac{\left(q^{-n};q\right)_k (b;q)_k}{(q;q)_k\,(c;q)_k} q^k$$

```
> convert(PS,qphihypergeom);
```

$$qphihypergeom\left([q^{-n}, b], [c], q, q\right)$$

9. Remarks

This article is a condensed version of the PhD thesis [Sprenger (2009)]. The *Maple* package `qFPS.mpl` together with some example files can be downloaded from the author's web site `http://www.mathematik.uni-kassel.de/~koepf/Publikationen`.

10. Acknowledgements

We would like to thank the anonymous referees for their very helpful comments and suggestions which improved the article considerably.

References

[Abramov, Paule, and Petkovšek (1998)] Abramov, S., Paule, P., Petkovšek, M., 1998. q-Hypergeometric solutions of q-difference equations. Discrete Math. 180, 3–22.

[Abramov, Petkovšek, and Ryabenko (2000)] Abramov, S., Petkovšek, M., Ryabenko, A., 2000. Special formal series solutions of linear operator equations. Discrete Math., 3–25.

[Böing and Koepf (1999)] Böing, H., Koepf, W., 1999. Algorithms for q-hypergeometric summation in computer algebra. J. Symbolic Comput. 28, 777–799.

[Cluzeau and van Hoeij (2005)] Cluzeau, T., van Hoeij, M., 2005. Computing hypergeometric solutions of linear recurrence equations. Appl. Algebra Engrg. Comm. Comput. 17, 83–115.

[Gasper and Rahman (1990)] Gasper, G., Rahman, M., 1990. Basic Hypergeometric Series. Vol. 35 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge.

[Horn (2008)] Horn, P., 2008. Faktorisierung in Schief-Polynomringen. Ph.D. thesis, Universität Kassel.

[Jackson (1909)] Jackson, F., 1909. q-form of Taylor's theorem. Messenger Math. 38, 62–64.

[Kac and Cheung (2002)] Kac, V., Cheung, P., 2002. Quantum Calculus. Springer, New York.

[Kauers and Koutschan (2009)] Kauers, M., Koutschan, C., 2009. A Mathematica package for q-holonomic sequences and power series. Ramanujan J. 19, 137–150.

[Koekoek and Swarttouw (1998)] Koekoek, R., Swarttouw, R. F., 1998. The Askey-scheme of Hypergeometric Orthogonal Polynomials and its q-Analogue. No. 98-17. Delft University of Technology.

[Koepf (1992)] Koepf, W., 1992. Power series in computer algebra. J. Symbolic Comput. 13, 581–603.

[Koepf (2003)] Koepf, W., 2003. Power series, Bieberbach conjecture and the de Branges and Weinstein functions. In: ISSAC 2003 Proceedings. ACM, pp. 169–175.

[Koepf, Rajković, and Marinković (2007)] Koepf, W., Rajković, P. M., Marinković, S. D., 2007. Properties of q-holonomic functions. J. Difference Equ. Appl. 13 (7), 621–638.

[Koepf and Schmersau (2001)] Koepf, W., Schmersau, D., 2001. On a structure formula for classical q-orthogonal polynomials. J. Comput. Appl. Math. (136), 99–107.

[Koornwinder (1993)] Koornwinder, T. H., 1993. On Zeilberger's algorithm and its q-analogue. J. Comput. Appl. Math. (48), 91–111.

[Koornwinder (1999)] Koornwinder, T. H., 1999. Some simple applications and variants of the q-binomial formula. Tech. rep., University of Amsterdam.

[Sprenger (2009)] Sprenger, T., 2009. Algorithmen für q-holonome Funktionen und q-hypergeometrische Reihen. Ph.D. thesis, Universität Kassel.

[van Hoeij (1998)] van Hoeij, M., 1998. Finite singularities and hypergeometric solutions of linear recurrence equations. J. Pure Appl. Algebra 139, 109–131.

[Zeilberger (1990)] Zeilberger, D., 1990. A holonomic systems approach to special function identities. J. Comput. Appl. Math. (32), 321–368.