J. Symbolic Computation (1999) **28**, 777–799
Article No. jsco.1998.0339
Available online at http://www.idealibrary.com on IDEAL

Algorithms for q-Hypergeometric Summation in Computer Algebra

HARALD BÖING[†] AND WOLFRAM KOEPF[‡]

[†]*Konrad-Zuse-Zentrum, Berlin, Germany*
[‡]*Hochschule für Technik, Wirtschaft und Kultur, Leipzig, Germany*

This paper describes three algorithms for q-hypergeometric summation:

- a multibasic analogue of Gosper's algorithm,
- the q-Zeilberger algorithm, and
- an algorithm for finding q-hypergeometric solutions of linear recurrences

together with their MAPLE implementations, which is relevant both to people being interested in symbolic computation and in q-series.

For all these algorithms, the theoretical background is already known and has been described, so we give only short descriptions, and concentrate ourselves on introducing our corresponding MAPLE implementations by examples. Each section is closed with a description of the input/output specifications of the corresponding MAPLE command.

We present applications to q-analogues of classical orthogonal polynomials. In particular, the connection coefficients between families of q-Askey–Wilson polynomials are computed.

MATHEMATICA implementations have been developed for most of these algorithms, whereas to our knowledge only Zeilberger's algorithm has been implemented in MAPLE so far (Koornwinder, 1993 or Zeilberger, cf. Petkovšek *et al.*, 1996).

We made an effort to implement the algorithms as efficient as possible which in the q-Petkovšek case led us to an approach with equivalence classes. Hence, our implementation is considerably faster than other ones. Furthermore the q-Gosper algorithm has been generalized to also find formal power series solutions.

1. Introduction

The well-known algorithms of Gosper, Zeilberger and Petkovšek are useful tools for problems dealing with hypergeometric summation. The theory and a description of their implementation in MAPLE are described in detail by Koepf (1998); the theory can also be found in Petkovšek *et al.* (1996); for MAPLE details see also Koepf (1996).

The algorithms can be easily adapted to the q-case, see e.g., Koornwinder (1993) for a short description of Gosper/Zeilberger and Abramov *et al.* (1998) for Petkovšek, and Gasper and Rahman (1990) as a general reference for q-hypergeometric (basic) series.

Hence we sketch the underlying theory only briefly and put emphasis on the application of our MAPLE package `qsum.mpl` which is an implementation of those algorithms. Our purpose is to show how these procedures can be applied to solve, e.g., problems related to orthogonal polynomials. Additionally we introduce a slight variation of the q-Gosper algorithm that searches for formal power series solutions and a multibasic version that is implemented in our package. To increase the usability of the programs we made an effort

0747–7171/99/120777 + 23 $30.00/0

to implement the algorithms as efficient as possible which in the q-Petkovšek case led us to an approach with equivalence classes. Hence, our implementation is considerably faster than other ones.

This package and an extensive help database for MAPLE V (Release 4 or 5) is available from the URL `http://www.imn.htwk-leipzig.de/~koepf/research.html` or by e-mail request from `koepf@imn.htwk-leipzig.de`. All examples shown in this paper were computed with MAPLE V Release 5 on a Pentium PC with 200 MHz. A corresponding MAPLE worksheet with the computations of this paper and the timings can also be obtained electronically from the above URL.

For MATHEMATICA the package `qZeil.m` written by Riese is an excellent implementation of q-Gosper's and q-Zeilberger's algorithm.[†] Petkovšek implemented the q-Petkovšek algorithm in MATHEMATICA.[‡] A Mathematica notebook with the examples of Table 2 and Section 4.1 using these packages is also available from the above URL.

Throughout this paper by $\boldsymbol{q}$ the vector $(q_1, q_2, \ldots, q_m)$, $m \in \mathbb{N}$ (set of positive integers) is denoted, and $\mathbb{F}$ is an abbreviation for $\mathbb{K}(q_1, q_2, \ldots, q_m)$, $\mathbb{K}$ denoting the field of rational numbers $\mathbb{Q}$ extended by some parameters.[§]

2. q-Gosper

The $\boldsymbol{q}$-version of Gosper's algorithm determines for a $\boldsymbol{q}$-*hypergeometric* term $\mathrm{F}(k)$, i.e.

$$\frac{\mathrm{F}(k+1)}{\mathrm{F}(k)} = \mathrm{H}(k)$$

for all $k \in \mathbb{Z}$ (set of integers) with

$$\mathrm{H}(k) \in \mathbb{F}\left(q_1^k, \ldots, q_m^k\right),$$

a $\boldsymbol{q}$-hypergeometric term $\mathrm{G}(k)$ with

$$\mathrm{F}(k) = \mathrm{G}(k+1) - \mathrm{G}(k) \tag{2.1}$$

iff it exists. Knowing such an *antidifference*[¶] (also called *indefinite sum*) $\mathrm{G}(k)$ for $\mathrm{F}(k)$ makes the evaluation of sums with given lower and upper bounds over $\mathrm{F}(k)$ easy:

$$\sum_{k=l}^{h} \mathrm{F}(k) = \mathrm{G}(h+1) - \mathrm{G}(l).$$

The $\boldsymbol{q}$-Gosper algorithm is based on the following two observations:

(1) By dividing equation (2.1) by $\mathrm{G}(k)$ one can see that $\mathrm{G}(k)/\mathrm{F}(k) \in \mathbb{F}\left(q_1^k, \ldots, q_m^k\right)$.
(2) On the other hand, if one divides equation (2.1) by $\mathrm{F}(k)$, one obtains

$$\frac{\mathrm{F}(k+1)}{\mathrm{F}(k)}\frac{\mathrm{G}(k+1)}{\mathrm{F}(k+1)} - \frac{\mathrm{G}(k)}{\mathrm{F}(k)} = 1,$$

[†]The package `qZeil` is available from `http://www.risc.uni-linz.ac.at/research/combinat/risc/software/qZeil`.

[‡]The corresponding package `qHyper.m` (and the hypergeometric version `Hyper.m`) can be obtained at `http://www.mat.uni-lj.si/dwnld.htm`.

[§]From a theoretical point of view any computable field of characteristic zero would be appropriate. However, algorithms used in computer algebra systems (such as factorization) usually require a more restrictive setting.

[¶]Note that by *antidifference* we always mean an *upward antidifference*.

or equivalently

$$\mathrm{H}(k)\,\mathrm{C}(k+1)-\mathrm{C}(k)=1,\qquad \mathrm{C}(k)=\frac{\mathrm{G}(k)}{\mathrm{F}(k)}\in\mathbb{F}\big(q_1^k,\dots,q_m^k\big)\,. \tag{2.2}$$

Thus to determine an antidifference $\mathrm{G}(k)$ for $\mathrm{F}(k)$ it is sufficient to search for a rational solution $\mathrm{C}(k)$ of an inhomogeneous recurrence equation of first order with rational coefficients.

As the q-Gosper algorithm is already well described by Koornwinder (1993) for the case $m=1$, by Riese (1996) for $m=2$, and by Böing (1998) for the general case $m\in\mathbb{N}$, we merely want to give the two main steps of the algorithm without going into detail:

(1) Determine $\mathrm{P}(k),\mathrm{Q}(k),\mathrm{R}(k)\in\mathbb{F}\big[q_1^k,\dots,q_m^k\big]$ such that

$$\mathrm{H}(k)=\frac{\mathrm{F}(k+1)}{\mathrm{F}(k)}=\frac{\mathrm{P}(k+1)}{\mathrm{P}(k)}\frac{\mathrm{Q}(k)}{\mathrm{R}(k)} \tag{2.3}$$

and

$$\gcd(\mathrm{Q}(k),\mathrm{R}(k+j))=1,\qquad \text{for all } j\in\mathbb{N}_0=\mathbb{N}\cup\{0\}. \tag{2.4}$$

(Note that the polynomials $\mathrm{P}(k),\mathrm{Q}(k),\mathrm{R}(k)$ are not uniquely determined by these two properties (cf. Lemma 4.1).)

(2) If the inhomogeneous first-order recurrence equation for $\mathrm{X}(k)$

$$\mathrm{Q}(k)\,\mathrm{X}(k)-\mathrm{R}(k-1)\,\mathrm{X}(k-1)=\mathrm{P}(k) \tag{2.5}$$

has a solution $\mathrm{X}(k)\in\mathbb{F}\big[q_1^k,q_1^{-k},\dots,q_m^k,q_m^{-k}\big]$, then

$$\mathrm{G}(k)=\frac{\mathrm{R}(k-1)\,\mathrm{X}(k-1)}{\mathrm{P}(k)}\,\mathrm{F}(k) \tag{2.6}$$

is a q-hypergeometric antidifference for $\mathrm{F}(k)$. Otherwise no such antidifference exists.

With the q-Gosper algorithm one can derive identities, like e.g., Formula (II.34) found in the Appendix of Gasper and Rahman (1990):

$$\sum_{k=0}^{n}\frac{1-a\,p^k\,q^k}{(1-a)\,c^k}\frac{(a;p)_k}{(a\,p/c;p)_k}\frac{(c;q)_k}{(q;q)_k}=\frac{1}{c^n}\frac{(a\,p;p)_n\,(c\,q;q)_n}{(a\,p/c;p)_n\,(q;q)_n}, \tag{2.7}$$

where $(a;q)_k$ denotes the *q-Pochhammer* symbol defined as usual by

$$(a;q)_k=\begin{cases}(1-a)(1-a\,q)\cdots(1-a\,q^{k-1}), & \text{if } k>0,\\ 1, & \text{if } k=0,\\ [(1-a\,q^{-1})(1-a\,q^{-2})\cdots(1-a\,q^{k})]^{-1}, & \text{if } k<0.\end{cases} \tag{2.8}$$

After loading our package in MAPLE via

```
> read `qsum.mpl`;
```

one can deduce the right-hand side of equation (2.7) from the left-hand side with the procedure `qgosper`:

```
> result:= qgosper((1-a*p^k*q^k)/(1-a)/c^k*qpochhammer(a,p,k)/
  qpochhammer(a*p/c,p,k)*qpochhammer(c,q,k)/qpochhammer(q,q,k),[p,q],k=0..n);
```

$$result := \mathrm{qpochhammer}(c,\, q,\, n+1)\,\mathrm{qpochhammer}(a,\, p,\, n+1)(-1+q^{(n+1)})$$
$$(-c+p^{(n+1)}\,a)c^{(-n-1)}\Big/($$
$$\mathrm{qpochhammer}(q,\, q,\, n+1)\mathrm{qpochhammer}\left(\frac{a\,p}{c},\, p,\, n+1\right)(-1+a)(-1+c)).$$

It is easy to check that this result is equivalent to the right-hand side of equation (2.7), e.g., by the computation

```
> qsimpcomb(result/qpochhammer(a*p,p,n)/qpochhammer(c*q,q,n)*
  qpochhammer(a*p/c,p,n)*qpochhammer(q,q,n)*c^n);
```

$$1$$

The procedure `qsimpcomb` is a simplification procedure for $\boldsymbol{q}$-hypergeometric terms. It was designed to simplify ratios of $\boldsymbol{q}$-hypergeometric terms like $\mathrm{F}(k+1)/\mathrm{F}(k)$ to decide rationality. If the result is nonrational, the procedure `qsimplify` returns a more compact result than `qsimpcomb`.

Generally the most time-consuming part in $\boldsymbol{q}$-Gosper's algorithm is its second step, which is usually done in two parts:†

(1) Determine lower and upper degree bounds l and h for $\mathrm{X}(k)$.
(2) Substitute a generic Laurent polynomial $\sum_{i=l}^{h} c_i\,(q^k)^i$ for $\mathrm{X}(k)$ with as yet unknown coefficients c_i in equation (2.5) and solve the resulting *linear* system by comparing the coefficients of $(q^k)^i$.

This method involves solving a linear system in $h-l+1$ variables which is inefficient if the difference $h-l$ is large. Abramov *et al.* (1995) introduced an alternative algorithm. It is based on the idea to convert the recurrence equation for $\mathrm{X}(k)$ into one for the coefficients c_i and using this recurrence to calculate as many coefficients as possible. Abramov *et al.* (1995) suggested using their algorithm if the difference $h-l$ is greater or equal to the order of the recurrence equation that is to be solved.

We want to illustrate this by calculating an antidifference for the function

$$\mathrm{F}_n(k) = \frac{(a;q)_k}{(q;q)_k}(a\,q^n)^{-k}, \tag{2.9}$$

where a is an arbitrary parameter. The application of q-Gosper's algorithm to $\mathrm{F}_n(k)$ with symbolic n shows that no q-hypergeometric antidifference exists:

†To simplify the notation, we just describe the case $m=1$.

```
> qgosper(qpochhammer(a,q,k)/qpochhammer(q,q,k)/(a*q^n)^k,q,k);

Error, (in qgosper) No q-hypergeometric antidifference exists.
```

However, for each nonnegative integer $n \in \mathbb{N}_0$ there exists an antidifference, where the resulting recurrence equation for $\mathrm{X}(k)$, i.e.

$$(1 - a\,q^k)\,\mathrm{X}(k) - a\,q^n(1 - q^k)\mathrm{X}(k-1) = 1, \tag{2.10}$$

implies $\mathrm{X}(k)$ being a polynomial of degree n. Thus for $n = 1$ we obtain, e.g.,footnoteBy default the resulting antidifference is factorized over $\mathbb{F}[q^k]$, which can be rather time-consuming if $\mathrm{X}(k)$ is 'complicated'. The application of this simplification can be dropped via the option `simplify=false`.

```
> qgosper(qpochhammer(a,q,k)/qpochhammer(q,q,k)/(a*q)^k,q,k,simplify=false);
```

$$\frac{a\,q\,(-1+q^k)\left(\dfrac{1}{-1+a\,q} + \dfrac{a\,(q-1)q^k}{(-1+a\,q)(-1+a)\,q}\right)\mathrm{qpochhammer}(a,q,k)}{(a\,q)^k\,\mathrm{qpochhammer}(q,q,k)}.$$

The system of linear equations can be solved by `qgosper`[†] with three different methods that are invoked via the optional argument "`solvemethod=`name", where name is one of

`ABP`: This represents the algorithm of Abramov *et al.* (1995) mentioned above that we implemented for $m = 1$.
`solve`: This is Maple's builtin `solve` procedure.
`gausselim`: An implementation of Gaussian elimination, since Maple's `solve` in many cases only poorly solves systems of linear equations. The same problem occurs in Mathematica and was described by Paule and Riese (1997).
`auto`: This is the default, meaning that `qgosper` chooses either `ABP` or `gausselim`, depending on the example. In most cases it chooses the faster method.

Table 1 shows the timings for computing antidifferences of $\mathrm{F}_n(k)$, generated by the call

```
qgosper(qpochhammer(a,q,k)/qpochhammer(q,q,k)/(a*q^n)^k,q,k,simplify=false);
```

with specified n, and the additional optional argument `solvemethod` set to the appropriate value:[‡]

In Section 4 we will introduce an extension of the one-dimensional q-Gosper algorithm that allows the computation of an antidifference for $\mathrm{F}_n(k)$ for arbitrary $n \in \mathbb{N}_0$.

Note that—similarly to the process of indefinite integration—an antidifference is only determined up to an additive constant, as the following example for $m = 3$ shows. The function `qbrackets(k,q)` is given as

$$[k]_q = \frac{q^k - 1}{q - 1}.$$

[†] This also applies to `qsumrecursion`, introduced in the next section.
[‡] The occurring dashes (—) in the table mean that no result was obtained within one hour.

Table 1. Timings for different solve methods.

n	10	20	30	40	50	60	70	80	90	100
`solve`	6 s	128 s	—	—	—	—	—	—	—	—
`gausselim`	1 s	3 s	8 s	19 s	39 s	76 s	136 s	210 s	294 s	425 s
`ABP`	0 s	1 s	3 s	9 s	23 s	48 s	86 s	163 s	199 s	275 s

We define $\mathrm{t}(k)$ by

```
> t:= qbrackets(k,p)*qbrackets(k,q)*qbrackets(k,r):
```

Obviously $\mathrm{t}(k)$ is an antidifference of $\mathrm{t}(k+1)-\mathrm{t}(k)$. Let us nevertheless compute an antidifference of $\mathrm{t}(k+1)-\mathrm{t}(k)$ applying `qgosper`:

```
> result:= qsimpcomb(qgosper(subs(k=k+1,t)-t,[p,q,r],k));
```

$$result := \frac{r^k + q^k - q^k\,r^k + p^k - p^k\,r^k - p^k\,q^k + p^k\,q^k\,r^k}{(p-1)(q-1)(r-1)}.$$

With `qsimpcomb` one shows that this antidifference is the same as `t` with an additive constant (w.r.t. k):

```
> qsimpcomb(result-t);
```

$$\frac{1}{(p-1)(q-1)(r-1)}.$$

2.1. DESCRIPTION OF THE MAPLE PROCEDURE `qgosper`

A thorough reference of the procedures in the package `qsum.mpl` is accessible in MAPLE V (Release 4 or 5) via the accompanying help file `maple.hdb`.[†] For a general introduction to the package one should type `?qsum` in MAPLE. To obtain help for a special procedure one can either follow the given links or use e.g., `?qgosper` directly in MAPLE.

Some of the procedures allow certain optional arguments. These can be specified in an arbitrary order after the regular arguments and are always of the type *optname*=*optvalue* where *optname* is the name of the option and *optvalue* a possible value for this option.

Calling Sequence:

```
qgosper(f,q,k,...);
qgosper(f,[q1,q2,...],k,...);
qgosper(f,q,k=l..h,...);
```

[†]To have access to the help database, set `libname:=‘libname‘,‘directory‘`. Here `directory` denotes the directory where `maple.hdb` resides.

Parameters:

`f`	— an algebraic expression, typically a product/ratio of rational functions, powers, `qpochhammer`'s, `qbinomial`'s, `qfactorial`'s, `qbrackets`' and `qphihyperterm`'s, compare Section 3.
`k`	— a name, the summation variable
`q,q1,q2,...`	— names
`l,h`	— algebraic expressions, the summation range
`...`	— a sequence of optional arguments (of type name=name)

Input/Output Description:

This function decides whether for a ***q***-hypergeometric term f(k) a ***q***-hypergeometric antidifference g(k) exists. If such an antidifference exists, it will be returned; otherwise an error message is generated stating that no such antidifference exists. If the input is not ***q***-hypergeometric w.r.t. k, the function will return an appropriate error message.

Options and Remarks:

- If `qgosper` is called by `qgosper(f,q,k=l..h)`, it will return a term that is equal to `sum(f,k=l..h)` if a ***q***-hypergeometric antidifference for `f` exists.
- If `qgosper` returns no antidifference, one might try the option `series=true`, thus allowing `qgosper` to search for an antidifference that is a q-hypergeometric series, see Section 4.3.
- By default `qgosper` returns an upward antidifference. This can be changed by using the option `antidifference=down`.
- Different solve methods can be accessed via the option `solvemethod=`*method* where *method* is one of `ABP`, `gausselim`, or `solve`.
- If the resulting antidifference is complicated, it might be useful to avoid any simplifications; this can be achieved via `simplify=false`. By default the antidifference is factorized. If you want to apply *simpfunc* to the result instead, use `simplify=`*simpfunc*.

3. q-Zeilberger

Wilf and Zeilberger (1992) showed that Zeilberger's algorithm can be easily carried over to the q-case (see also the description by Koornwinder, 1993). The q-Zeilberger algorithm tries to derive a recurrence equation for the *definite sum* S(n)

$$\mathrm{S}(n) = \sum_{k=a_n}^{b_n} \mathrm{F}(n,k), \tag{3.1}$$

where F(n,k) is a q-hypergeometric term w.r.t. n and k. It uses the q-Gosper algorithm to determine an antidifference G(n,k) for $\mathrm{f}_n(k)$ and $\sigma_0(n),\ldots,\sigma_J(n) \in \mathbb{F}(q^n)$ for some $J \in \mathbb{N}$, where[†]

$$\mathrm{f}_n(k) = \sum_{j=0}^{J} \sigma_j(n)\,\mathrm{F}(n-j,k). \tag{3.2}$$

[†] Instead of F($n-j,k$) one could also use F($n+j,k$) leading to an upward instead of a downward recurrence equation.

Note that q-Gosper's algorithm can be applied to $\mathrm{f}_n(k)$ since it is a rational multiple of $\mathrm{F}(n,k)$. The application of q-Gosper to $\mathrm{f}_n(k)$ in the second step gives a recurrence equation for $\mathrm{X}(k)$ (see equation (2.5)) which leads to a system of equations that are linear in the coefficients of $\mathrm{X}(k)$ *and* in $\sigma_0(n),\dots,\sigma_J(n)$. Thus the only change required is to add the unknowns $\sigma_0(n),\dots,\sigma_J(n)$ to the variables of the linear system that has to be solved.

If the algorithm is successful, we obtain an inhomogeneous recurrence equation for $\mathrm{F}(n,k)$,

$$\sum_{j=0}^{J}\sigma_j(n)\,\mathrm{F}(n-j,k)=\mathrm{G}(n,k+1)-\mathrm{G}(n,k), \tag{3.3}$$

with $\sigma_0(n),\dots,\sigma_J(n)\in\mathbb{F}(q^n)$. By summing over k from $\alpha_n=\max\{a_{n-j}\,|\,j=0,\dots,J\}$ to $\beta_n=\min\{b_{n-j}\,|\,j=0,\dots,J\}$ we get the inhomogeneous recurrence equation

$$\sum_{j=0}^{J}\sigma_j(n)\,\mathrm{S}(n-j)=\mathrm{G}(n,\beta_n+1)-\mathrm{G}(n,\alpha_n)+\sum_{j=0}^{J}\left[\sum_{k=a_{n-j}}^{\alpha_n-1}\mathrm{F}(n-j,k)+\sum_{k=\beta_n+1}^{b_{n-j}}\mathrm{F}(n-j,k)\right] \tag{3.4}$$

for $S(n)$.

An advantage of Zeilberger's method is that, no matter how complicated the *computation* of the recurrence equation (3.4) might be, one can *prove* its validity by rational arithmetic if one also outputs $\mathrm{G}(n,k)$ or the ratio $\mathrm{G}(n,k)/\mathrm{F}(n,k)$:

(1) Write down the corresponding recurrence equation (3.3) for the summand $\mathrm{F}(n,k)$ and divide it by $\mathrm{F}(n,k)$. As $\mathrm{G}(n,k)/\mathrm{F}(n,k)\in\mathbb{F}\big(q^n,q^k\big)$, this equation can be shown by pure *rational arithmetic*.†
(2) Knowing that equation (3.3) is correct, we proceed by summing over k to obtain equation (3.4).

Thus the term $\mathrm{G}(n,k)/\mathrm{F}(n,k)$ is often referred to as the (rational) *certificate* for the recurrence equation (3.4). Since the term $\mathrm{F}(n,k)$ usually has *finite support*, i.e. $\mathrm{F}(n,k)$ vanishes for each $n\in\mathbb{N}_0$ outside a finite interval of k, in most cases the right-hand side of (3.4) is zero. Therefore our implementation assumes that the right-hand side of (3.4) is zero, if no summation bounds are specified (thus saving computation time).
Suppose we want to prove the q-Pfaff–Saalschütz identity

$${}_3\phi_2\left(\begin{matrix}a,b,q^{-n}\\ c,a\,b\,q^{1-n}/c\end{matrix}\,\middle|\,q;\,q\right)=\frac{(c/a;q)_n\,(c/b;q)_n}{(c;q)_n\,(c/(a\,b);q)_n},\qquad(n\in\mathbb{N}_0) \tag{3.5}$$

where the *basic hypergeometric series* ${}_r\phi_s$ is defined by

$${}_r\phi_s\left(\begin{matrix}a_1,a_2,\dots,a_r\\ b_1,b_2,\dots,b_s\end{matrix}\,\middle|\,q;\,z\right)=\sum_{k=0}^{\infty}\frac{(a_1,a_2,\dots,a_r;q)_k}{(b_1,b_2,\dots,b_s;q)_k}\,\frac{z^k}{(q;q)_k}\Big((-1)^k\,q^{\binom{k}{2}}\Big)^{1+s-r}, \tag{3.6}$$

and $(a_1,a_2,\dots,a_r;q)_k$ is an abbreviation for $\prod_{j=1}^{r}(a_j;q)_k$. With our implementation of the q-Zeilberger algorithm (`qsumrecursion`) one obtains:

†After simplifying the ratios of the q-hypergeometric terms.

```
> qsumrecursion([a,b,q^(-n)],[c,a*b*q^(1-n)/c],q,q,S(n),
  rec2qhyper=true,sumrange=0..n);
```

$$\left[\mathrm{S}(n) = \frac{\mathrm{qpochhammer}\left(\frac{c}{a}, q, n\right)\ \mathrm{qpochhammer}\left(\frac{c}{b}, q, n\right)}{\mathrm{qpochhammer}(c, q, n)\ \mathrm{qpochhammer}\left(\frac{c}{b\,a}, q, n\right)},\ 0 \le n \right].$$

To enforce the computation of the inhomogeneous part† we supplied the optional argument `sumrange=0..n`; the option `rec2qhyper=true` advises `qsumrecursion` to return a q-hypergeometric term instead of a recurrence equation if the order is one.

Additionally to the output, the procedure stored some information in the global variable `_qsumrecursion_proof` which is a table with several entries. One obtains, e.g., the input term, the recurrence equation and the certificate $\mathrm{G}(n,k)/\mathrm{F}(n,k)$ for (3.3) by

```
> term:= _qsumrecursion_proof[_F];
```

$$term := \frac{\mathrm{qpochhammer}(a, q, k)\ \mathrm{qpochhammer}(b, q, k)\ \mathrm{qpochhammer}\left(q^{-n}, q, k\right)\ q^{k}}{\mathrm{qpochhammer}(c, q, k)\ \mathrm{qpochhammer}\left(\frac{a\,b\,q^{1-n}}{c}, q, k\right)\ \mathrm{qpochhammer}(q, q, k)}$$

```
> RE:= _qsumrecursion_proof[_recursion];
```

$$RE := (q^n\,c - q)\,(-q^n\,c + a\,b\,q)\mathrm{S}(n) + (-q^n\,c + b\,q)\,(a\,q - q^n\,c)\,\mathrm{S}(n-1) = 0$$

```
> cert:= _qsumrecursion_proof[_certificate];
```

$$cert := -\frac{(-q^n\,c + a\,b\,q)\,(-q + c\,q^k)\,(-1 + q^k)\,q^n}{(q^n - 1)\,q^k}.$$

Due to the ${}_r\phi_s$-series type of input `qsumrecursion` had to choose a summation variable (namely k) such that $\mathrm{S}(n)$ is a sum of `term` over k from 0 to n. However, be careful to avoid interference with the global variable k we defined k to be local.‡ The best way to use this local variable (stored in `_qsumrecursion_proof[_sumvar]`) is to assign it to a global variable, e.g.

```
> i:=_qsumrecursion_proof[_sumvar];
```

$$i := k$$

One can now prove the recurrence equation (3.3) for the summand $\mathrm{F}(n,k)$ (`term`) by rational arithmetic, where the antidifference $\mathrm{G}(n,k)$ is $\mathrm{F}(n,k)\frac{\mathrm{G}(n,k)}{\mathrm{F}(n,k)}$, hence `cert*term`:

† As we expected, it turned out to be zero.

‡ In other words, a substitution, such as `subs(k=k+1,term)` would *not* work!

```
> result:=subs({S(n-1)=subs(n=n-1,term),S(n)=term},lhs(RE))-
      (subs(i=i+1,cert*term)-cert*term):
> qsimpcomb(result);
```

$$0$$

Therefore equation (3.3) is valid, which can now be converted into a recurrence equation for $\mathrm{S}(n)$ by summing over k: because $\mathrm{F}(n,k)$ and the antidifference $\mathrm{G}(n,k)$ have *finite support*, i.e.

$$\frac{(a;q)_k\,(b;q)_k\,(q^{-n};q)_k\,q^k}{(c;q)_k\,(a\,b\,q^{1-n}/c;q)_k\,(q;q)_k}=0,\qquad \text{for } k<0 \text{ or } k>n$$

the easiest way to prove `_qsumrecursion_proof[_recursion]` now is by summing over k from $-\infty$ to ∞, showing that the inhomogeneous part is zero.

Next let us apply our program to deduce the right-hand side of the q-Dixon identity

$$\sum_{k=-n}^{n}(-1)^k\,q^{k\,(3\,k-1)/2}\begin{bmatrix}n+b\\n+k\end{bmatrix}_q\begin{bmatrix}n+c\\c+k\end{bmatrix}_q\begin{bmatrix}b+c\\b+k\end{bmatrix}_q=\frac{(q;q)_{n+b+c}}{(q;q)_n\,(q;q)_b\,(q;q)_c},\qquad(3.7)$$

where the q-binomial coefficient is defined by

$$\begin{bmatrix}n\\k\end{bmatrix}_q=\frac{(q;q)_n}{(q;q)_k\,(q;q)_{n-k}},\qquad(n\in\mathbb{N}_0).$$

First we note that the command `sum2qhyper` can be used to convert q-hypergeometric series into q-hypergeometric notation, hereby unmasking their lower and upper parameters as well as their argument; we obtain e.g., for the q-Dixon sum:

```
> term:= (-1)^k*q^(k*(3*k-1)/2)*qbinomial(n+b,n+k,q)*qbinomial(n+c,c+k,q)*
  qbinomial(b+c,b+k,q):
> assume(n,integer):
  sum2qhyper(term,q,k);
  n:='n':
```

$$(-1)^{(-n)}\,q^{(1/2\,n+3/2\,n^2)}\,\mathrm{qpochhammer}(q,\,q,\,b+c)$$
$$\phi([q^{(-2\,n)},\,q^{(-n-b)},\,q^{(-c-n)}],\,[q^{(1-n+b)},\,q^{(1-n+c)}],\,q,\,q^{n+b+c+1})/($$
$$\mathrm{qpochhammer}(q,\,q,\,c-n)\,\mathrm{qpochhammer}(q,\,q,\,2\,n)\,\mathrm{qpochhammer}(q,\,q,\,b-n)).$$

The assumption that n is an integer was made so that the program can compute the appropriate shift to let the series start with $k=0$. The upper parameters of the resulting expression show in particular that the sum consists of $2n+1$ summands (if neither b nor c are integers smaller than n).

If we apply `qsumrecursion` to find a recurrence equation for the sum, we obtain

```
> qDixon_RE:= qsumrecursion(term,q,k,S(n));
```

$$
\begin{aligned}
qDixon\_RE := {} & \left(-q^{2n}+q\right)\left(q^n+1\right)\left(q^n-1\right)q^3\,\mathrm{S}(n)+q\left(q^5+q^4-q^{4+n+c+b}-\right.\\
& q^{2n+3}-q^{n+3+c}-q^{n+3+c+b}+q^3+q^{2n+3+c}+q^{2n+3+b}-q^{n+3+b}-q^{3n+2}+\\
& q^{2n+2+b}+q^{4n+2+c+b}+q^{2n+2+c}-q^{3n+2+b}-q^{3n+2+c}-q^{2n+2+c+b}-q^{1+3n}+\\
& \left.q^{4n+1+c+b}+q^{4n+c+b}\right)\mathrm{S}(n-1)-\left(q-q^{n+c+b}\right)\left(-q^{3n+c+b+1}+q^5+q^{2n+2}-\right.\\
& q^{3n+c+b}+q^4-q^{3n+2+c+b}+q^6-q^{4+n+c+b}+q^{2n+3+b}-q^{n+3+c}+q^{2n+3+c}-\\
& \left.q^{n+3+b}-q^{4+n+c}+q^{2n+2+b}+q^{2n+2+c}-q^{4+n+b}\right)\mathrm{S}(n-2)+\\
& \left(q^2-q^{n+c+b}\right)\left(q-q^{n+c+b}\right)\left(q^2-q^{n+b}\right)\left(q^2-q^{n+c}\right)\mathrm{S}(n-3) \quad = \quad 0, \qquad (3.8)
\end{aligned}
$$

which is definitely not the result we want. Here we have an example (of quite a few), where the q-Zeilberger algorithm does not find a recurrence equation of *minimal* order, which was pointed out by Paule and Riese (1997). Paule (1994) introduced the method of *creative symmetrizing*—a generalization of which is given by the following lemma—resolving the problem of non-minimality in most cases.

LEMMA 3.1. (RIESE, 1995) *If for some* $c \in \mathbb{Z}$

$$\sum_{k=a_n}^{b_n} \mathrm{F}(n,k) = \sum_{k=a_n}^{b_n} \mathrm{F}(n,-k-c),$$

and $\mathrm{M}(n,k) = \mathrm{F}(n,-k-c)/\mathrm{F}(n,k)$ *then*

$$\sum_{k=a_n}^{b_n} \mathrm{F}(n,k) = \frac{1}{2}\sum_{k=a_n}^{b_n} \left(1+\mathrm{M}(n,k)\right)\mathrm{F}(n,k).$$

The astonishing fact is that, in most cases, the q-Zeilberger algorithm applied to the symmetrized summand $1/2\,(1+\mathrm{M}(n,k))\,\mathrm{F}(n,k)$ increases the chance of getting a recurrence equation of minimal order. For q-Dixon we choose $c=0$ to obtain:

```
> M:= qsimpcomb(subs(k=-k,term)/term);
```

$$M := q^k$$

If we now apply `qsumrecursion` to the symmetrized summand, we obtain:

```
> qsumrecursion((1+M)/2*term,q,k,S(n));
```

$$\left(q^n-1\right)\mathrm{S}(n) \; - \; \left(-1+q^{n+b+c}\right)\mathrm{S}(n-1) = 0.$$

Table 2 is an extract of the one given in Paule and Riese (1997) with the most time-consuming examples, where we used the newest version of their package `qZeil` (Version 1.8) with MATHEMATICA 3.0.

Turbo-`qZeil` just means that infinite summation bounds were specified, thus disabling the computation of the inhomogeneous part. Our procedure did not compute the inhomogeneous part either, but instead of specifying the order of the recurrence (which is

Table 2. Comparison of the procedures `qsumrecursion` and `qZeil`.

Equation†		Order	`qsumrecursion`	Turbo-`qZeil`
(III.18)	lhs	2	2 s	4 s
	rhs	2	1 s	5 s
(III.25)	lhs	3	4 s	11 s
	rhs	3	13 s	27 s
(III.28)	lhs	2	17 s	31 s
	rhs	2	20 s	39 s

necessary for `qZeil`), `qsumrecursion` tries to find a recurrence from first order up to the one, where it is successful. By default it looks for a recurrence of order one up to five, as in most cases the computing time of the non-successful tries is not crucial.

Note that an earlier implementation of `qZeil` failed with some of these examples, see Paule and Riese (1997).

The q-Zeilberger algorithm also gives a very simple method to deduce recurrence equations for orthogonal polynomials from the basic Askey–Wilson tableau (see e.g. Koekoek and Swarttouw, 1994). For the little q-Legendre polynomials

$$p_n(x|q) = {}_2\phi_1\left(\begin{matrix} q^{-n}, q^{n+1} \\ q \end{matrix} \,\middle|\, q;\, qx\right)$$

we get e.g., the recurrence equations

```
> qsumrecursion([q^(-n),q^(n+1)],[q],q,q*x,p(n));
```

$$q^n\,(q^n-1)\,(q+q^n)\,\mathrm{p}(n) + (2\,q^n - x\,q^{n+1} - q\,x - x\,q^n - x\,q^{2\,n})\,(q-q^{2\,n})\,\mathrm{p}(n-1) - q^n\,(q^n+1)\,(q-q^n)\,\mathrm{p}(n-2) = 0$$

with respect to n, and

```
> qsumrecursion([q^(-n),q^(n+1)],[q],q,q*q^y,p(y));
```

$$q^{n+1}\,(q^y-1)\,\mathrm{p}(y) \;-\; (q^y + q^{y+2\,n+1} - 2\,q^{n+1})\,\mathrm{p}(y-1) \;+\; q^n\,(q^y-q)\mathrm{p}(y-2) = 0$$

with respect to y. One can also compute the *q-difference equation* for the little q-Legendre polynomials

```
> qsumdiffeq([q^(-n),q^(n+1)],[q],q,q*x,p(x));
```

$$q\,(q^n-1)\,(q^n\,q-1)\,\mathrm{p}(x) + (-1+q)\,(-q^n + q^n\,q - q^3\,x\,q^n + q^2\,x\,(q^n)^2 + q\,x - q^2\,x\,q^n)\cdot Dq_x(\mathrm{p}(x)) - q^n\,(-1+q^2\,x)\,x\,(-1+q)^2\,q\,Dq_{x,\,x}(\mathrm{p}(x)) = 0,$$

†All equation numbers refer to the Appendix of the book of Gasper and Rahman (1990).

Dq_x denoting the *q-derivative operator*

$$Dq_x\, f(x) := \frac{f(x) - f(qx)}{(1-q)x}.$$

Furthermore, the computation

```
> qsumrecursion([q^(-n),q^(n+1)],[q],q,q,p(n),rec2qhyper=true);
```

$$\left[\mathrm{p}(n) = (-q)^n\, q^{\mathrm{binomial}(n,2)},\ 0 \le n\right]$$

shows that

$$p_n(1|q) = (-1)^n\, q^{n\,(n+1)/2}.$$

Now consider the Al–Salam–Chihara polynomials with $x = \cos\theta$, given by:

$$Q_n(x; a, b|q) = \frac{(a\,b;q)_n}{a^n}\, {}_3\phi_2\left(\begin{matrix} q^{-n}, a\,e^{i\,\theta}, a\,e^{-i\,\theta} \\ a\,b, 0 \end{matrix}\,\middle|\, q;\ q\right) \tag{3.9}$$

$$= \left(b\,e^{-i\,\theta}; q\right)_n\, e^{i\,n\,\theta}\, {}_2\phi_1\left(\begin{matrix} q^{-n}, a\,e^{i\,\theta} \\ b^{-1}\,q^{1-n}\,e^{i\,\theta} \end{matrix}\,\middle|\, q;\ b^{-1}\,q\,e^{-i\,\theta}\right). \tag{3.10}$$

With `qsumrecursion` one can easily show the equality of the two representations by

```
> re1:= qsumrecursion(qpochhammer(a*b,q,n)/a^n,
  [q^(-n),a*exp(I*theta),a*exp(-I*theta)],[a*b,0],q,q,Q(n));
```

$$-e^{I\,\theta}\,q^3\,\mathrm{Q}(n) \;-\; \left(e^{I\,\theta}\,q^n\,a - q + e^{I\,\theta}\,q^n\,b - q\,e^{2\,I\,\theta}\right) q^2\,\mathrm{Q}(n-1) \;+\; e^{I\,\theta}\left(-q^2 + a\,b\,q^n\right)\left(-q^n + q\right)\mathrm{Q}(n-2) = 0$$

```
> re2:= qsumrecursion(qpochhammer(b*exp(-I*theta),q,n)*exp(I*n*theta),
  [q^(-n),a*exp(I*theta)],[b^(-1)*q^(1-n)*exp(I*theta)],
  q,b^(-1)*q*exp(-I*theta),Q(n));
```

$$-e^{I\,\theta}\,q^3\,\mathrm{Q}(n) \;-\; \left(e^{I\,\theta}\,q^n\,a - q + e^{I\,\theta}\,q^n\,b - q\,e^{2\,I\,\theta}\right) q^2\,\mathrm{Q}(n-1) \;+\; e^{I\,\theta}\left(-q^2 + a\,b\,q^n\right)\left(-q^n + q\right)\mathrm{Q}(n-2) = 0.$$

Since the two recurrence equations agree, (3.9) and (3.10) really define the same polynomials as long as the two initial values $Q_0(x; a, b|q)$ and $Q_1(x; a, b|q)$ are shown to be equal for both representations.†

Finally we consider the Askey–Wilson polynomials $p_n(x; a, b, c, d|q)$ with $x = \cos\theta$, given by

$$p_n(x; a, b, c, d|q) = (a\,b, a\,c, a\,d; q)_n\ a^{-n}$$

† Note that the Riese implementation does not allow this type of input and returns the error message: *$a\,e^{i\,\theta}$ is not a valid power product.* With this package you have to replace $e^{i\,\theta}$ by y, apply the procedure `qZeil`, and use the backsubstitution $y = e^{i\,\theta}$ and the `Simplify` command.

$$ {}_4\phi_3\left(\left.\begin{matrix} q^{-n}, a\,b\,c\,d\,q^{n-1}, a\,e^{i\,\theta}, a\,e^{-i\,\theta} \\ a\,b, a\,c, a\,d \end{matrix}\right| q;\, q\right) \tag{3.11} $$

which are on top of the basic Askey–Wilson tableau (see e.g. Koekoek and Swarttouw, 1994). All the other basic orthogonal polynomial families in this tableau can be constructed from them by limiting procedures.

Gasper and Rahman (1990, Sections 7.5 and 7.6), considered the *connection coefficients* $c_k(n)$ between two Askey-Wilson polynomials with different parameters

$$ p_n(x;\alpha,\beta,\gamma,d|q) = \sum_{k=0}^{n} c_k(n;\alpha,\beta,\gamma,a,b,c,d)\, p_k(x;a,b,c,d|q). \tag{3.12} $$

This equation expresses the polynomials of one family by a linear combination of polynomials of another family.

Askey and Wilson (1985) showed the following representation for $c_k(n)$

$$ \begin{aligned} c_k(n;\alpha,\beta,\gamma,a,b,c,d) = {} & \frac{(\alpha\,d,\beta\,d,\gamma\,d,q;q)_n\ \left(\alpha\,\beta\,\gamma\,d\,q^{n-1};q\right)_k}{(\alpha\,d,\beta\,d,\gamma\,d,q,a\,b\,c\,d\,q^{k-1};q)_k\ (q;q)_{n-k}}\, q^{k^2-n\,k}\, d^{k-n} \\ & {}_5\phi_4\left(\left.\begin{matrix} q^{k-n}, \alpha\,\beta\,\gamma\,d\,q^{n+k-1}, a\,d\,q^k, b\,d\,q^k, c\,d\,q^k \\ a\,b\,c\,d\,q^{2\,k}, \alpha\,d\,q^k, \beta\,d\,q^k, \gamma\,d\,q^k \end{matrix}\right| q;\, q\right) \end{aligned} \tag{3.13} $$

in terms of a basic hypergeometric function.

For the special case $\beta = b$ and $\gamma = c$ Gasper and Rahman (1990) gave the formula

$$ c_k(n;\alpha,b,c,a,b,c,d) = \frac{(\alpha/a;q)_{n-k}\ \left(\alpha\,b\,c\,d\,q^{n-1};q\right)_k\,(q,b\,c,b\,d,c\,d;q)_n\ a^{n-k}}{(q,b\,c,b\,d,c\,d;q)_k\ (a\,b\,c\,d\,q^{k-1};q)_k\ (q,a\,b\,c\,d\,q^{2\,k};q)_{n-k}} \tag{3.14} $$

as a q-hypergeometric term.

Note that in their book this equation—which is number (7.6.9)—contains a misprint, discovered by our software, i.e. the factor $(q;q)_n$ is missing (cf. Gasper and Rahman, 1997).

From equation (3.13) with $\beta = b$ and $\gamma = c$ our implementation detects a recurrence equation of first order for $c_k(n)$, from which the resulting q-hypergeometric term is computed using an initial value for $k = n$:

```
> c:=(k,n,alpha,beta,gamma,a,b,c,d,j)->
  qpochhammer(alpha*d,beta*d,gamma*d,q,q,n)*
  qpochhammer(alpha*beta*gamma*d*q^(n-1),q,k)/
  qpochhammer(alpha*d,beta*d,gamma*d,q,a*b*c*d*q^(k-1),q,k)/
  qpochhammer(q,q,n-k)*q^(k^2-n*k)*d^(k-n)*
  qphihyperterm([q^(k-n),alpha*beta*gamma*d*q^(n+k-1),a*d*q^k,b*d*q^k,c*d*q^k],
  [a*b*c*d*q^(2*k),alpha*d*q^k,beta*d*q^k,gamma*d*q^k],q,q,j):
> intermediate:=qsumrecursion(c(k,N+k,alpha,b,c,a,b,c,d,j),q,j,S(N),
  rec2qhyper=true,sumrange=0..N):
  final:= subs(N=n-k,rhs(intermediate[1]));
```

$$ \begin{aligned} \mathit{final} := {} & ((-q + b\,d\,q^k\,\alpha\,c)\,\mathrm{qpochhammer}(b\,d\,q^k\,\alpha\,c,\, q,\, k)\,(-q + a\,b\,c\,d\,(q^k)^2) \\ & \mathrm{qpochhammer}(b\,c\,q^k,\, q,\, n-k)\,\mathrm{qpochhammer}\left(\frac{\alpha\,b\,c\,d\,(q^k)^2}{q},\, q,\, n-k\right) \end{aligned} $$

$$\mathrm{qpochhammer}\left(\frac{\alpha}{a},\, q,\, n-k\right)\mathrm{qpochhammer}(q\,q^k,\, q,\, n-k)$$
$$\mathrm{qpochhammer}(c\,d\,q^k,\, q,\, n-k)\,\mathrm{qpochhammer}(b\,d\,q^k,\, q,\, n-k)\,a^{(n-k)}\Big)\Big/($$
$$(-q+\alpha\,b\,c\,d\,(q^k)^2)\,(-q+a\,b\,c\,d\,q^k)\,\mathrm{qpochhammer}(a\,b\,c\,d\,q^k,\, q,\, k)$$
$$\mathrm{qpochhammer}(q,\, q,\, n-k)\,\mathrm{qpochhammer}(a\,b\,c\,d\,(q^k)^2,\, q,\, n-k)$$
$$\mathrm{qpochhammer}\left(\frac{b\,d\,q^k\,\alpha\,c}{q},\, q,\, n-k)\right)$$

which gives the right-hand side of equation (3.14).† This can be easily checked by the computation

```
> RHS:=qpochhammer(alpha/a,q,n-k)*qpochhammer(alpha*b*c*d*q^(n-1),q,k)*
  qpochhammer(b*c,b*d,c*d,q,n)*a^(n-k)*qpochhammer(q,q,n)/
  (qpochhammer(q,b*c,b*d,c*d,q,k)*qpochhammer(a*b*c*d*q^(k-1),q,k)*
  qpochhammer(q,a*b*c*d*q^(2*k),q,n-k)):
  qsimpcomb(final/RHS);
```

1

Note that similar computations with just one differing parameter—i.e. $\alpha = a, \gamma = c$ or $\alpha = a, \beta = b$—also lead to recurrence equations of order one and hence to q-hypergeometric terms. These results are given by

$$c_k(n;a,\beta,c,d,a,b,c,d) = \frac{(-q+d\,\beta\,c\,q^k\,a)\,(-q+a\,b\,c\,d\,q^{2k})\,b^{n-k}}{(-q+d\,\beta\,c\,q^{2k}\,a)\,(-q+a\,b\,c\,d\,q^k)}\,\frac{\left(d\,\beta\,c\,q^k\,a;q\right)_k}{\left(a\,b\,c\,d\,q^k;q\right)_k}$$
$$\times\frac{\left(a\,c\,q^k, d\,\beta\,c\,q^{2k-1}\,a, \beta/b, q^{k+1}, c\,d\,q^k, a\,d\,q^k;q\right)_{n-k}}{\left(q, a\,b\,c\,d\,q^{2k}, d\,\beta\,c\,q^{k-1}\,a;q\right)_{n-k}}$$

and

$$c_k(n;a,b,\gamma,d,a,b,c,d) = \frac{(-q+d\,\gamma\,q^k\,a\,b)\,(-q+a\,b\,c\,d\,q^{2k})\,c^{n-k}}{(-q+a\,b\,\gamma\,d\,q^{2k})\,(-q+a\,b\,c\,d\,q^k)}\,\frac{\left(d\,\gamma\,q^k\,a\,b;q\right)_k}{\left(a\,b\,c\,d\,q^k;q\right)_k}$$
$$\times\frac{\left(a\,b\,q^k, a\,b\,\gamma\,d\,q^{2k-1}, \gamma/c, q^{k+1}, b\,d\,q^k, a\,d\,q^k;q\right)_{n-k}}{\left(q, a\,b\,c\,d\,q^{2k}, d\,\gamma\,q^{k-1}\,a\,b;q\right)_{n-k}}$$

from which one can derive the connection coefficients between many families of the q-Askey–Wilson tableau by limit computations. By the symmetry of equation (3.11), a similar connection is valid if the fourth parameter d is varied.

3.1. DESCRIPTION OF THE MAPLE PROCEDURE `qsumrecursion`

Calling Sequence:

```
qsumrecursion(f,q,k,s(n),...);
qsumrecursion(f,q,k=l..h,s(n),...);
```

† The procedure `qphihyperterm` returns the summand of the basic hypergeometric series (3.6).

```
qsumrecursion(f,upper,lower,qq,z,s(n),...);
```

Parameters:

`f`	— an algebraic expression
`q`	— a name
`k`	— a name, the summation variable
`s`	— a name, the recurrence function
`n`	— a name, the recurrence variable
`l,h`	— algebraic expressions, lower and upper summation bounds
`upper, lower`	— lists of algebraic expressions, the upper and lower parameters of the general q-hypergeometric function
`qq`	— a name or a name$^{\text{integer}}$
`...`	— a sequence of optional arguments (of type name=name)

Input/Output Description:

`qsumrecursion` applies the q-Zeilberger algorithm to determine a recurrence equation for the sum $\mathrm{s}(n) = \sum_{k=l}^{h} \mathrm{f}(n,k)$. If successful, the recurrence equation will be returned; otherwise an error message is generated.

Options and Remarks:

- The last calling sequence is a shortcut for
  `qsumrecursion(f*qphihyperterm(upper,lower,qq,z,j),q,j,s(n));`
  where the prefactor `f` can be omitted if it is equal to one.
- The option `recursion` may be set to `up` or `down`, instructing `qsumrecursion` to return an upward or downward recurrence equation, respectively.
- Similarly as in `qgosper` the option `solvemethod` may be set.
- If no explicit summation range is given, then it is assumed that the bounds are `-infinity..infinity`, and a homogeneous recurrence equation is computed. If other bounds are given, then `qsumrecursion` computes the inhomogeneous part of the recurrence.
- Additionally one can supply the argument `inhomo2homo=true`; then `qsumrecursion` will convert a resulting inhomogeneous recurrence equation of order J into a homogeneous one of order $J+1$ if possible.
- By default `qsumrecursion` calculates an upward certificate for the recurrence equation; this may be changed by `certificate=down`.
- The option `rec2qhyper=true` tells the procedure to return a list with two entries if a recurrence equation of first order was found: the first entry is a q-hypergeometric term equal to the sum $\mathrm{s}(n)$, the second entry a restriction on n.
- The option `proof` can be set to any (unassigned) name. Then all necessary information needed to prove the recurrence equation will be stored as a table in that name; by default this information is stored in the global variable `_qsumrecursion_proof`.

4. *q*-Petkovšek

Abramov *et al.* (1998) gave a few algorithms to find solutions of special types of homogeneous and inhomogeneous recurrence equations with polynomial coefficients. Petkovšek implemented the procedure `qHyper` in MATHEMATICA that finds q-hypergeometric solutions of homogeneous recurrence equations.

4.1. *q*-HYPERGEOMETRIC SOLUTIONS OF HOMOGENEOUS RECURSIONS

Assume that we want to determine all q-hypergeometric term solutions $\mathrm{F}(n)$ of the homogeneous recurrence equation

$$\sum_{j=0}^{J} \sigma_j(n)\,\mathrm{F}(n+j) = 0, \qquad \sigma_0(n), \ldots, \sigma_J(n) \in \mathbb{F}[q^n]\,. \tag{4.1}$$

As in Gosper's algorithm (see equation (2.3)), the following *q-Gosper–Petkovšek representation* of rational functions plays a fundamental role:

LEMMA 4.1. *For every nonzero rational function* $\mathrm{L}(n) \in \mathbb{F}(q^n)$ *there exists a unique quadruple* $(z, \mathrm{P}(n), \mathrm{Q}(n), \mathrm{R}(n)) \in (\mathbb{F} \times \mathbb{F}[q^n]^3)$, *the* q-Gosper–Petkovšek representation *of* $\mathrm{L}(n)$, *with*

(i) $\displaystyle \mathrm{L}(n) = z\,\frac{\mathrm{P}(n+1)}{\mathrm{P}(n)}\,\frac{\mathrm{Q}(n)}{\mathrm{R}(n)}$,

(ii) $\gcd(\mathrm{Q}(n), \mathrm{R}(n+j)) = 1$, *for all* $j \in \mathbb{N}_0$,

(iii) $\gcd(\mathrm{Q}(n), \mathrm{P}(n)) = 1$, *and* $\gcd(\mathrm{R}(n), \mathrm{P}(n+1)) = 1$,

(iv) $\mathrm{P}(n), \mathrm{Q}(n)$, *and* $\mathrm{R}(n)$ *are monic,*[†] *and* $\operatorname{ldeg}(\mathrm{P}(n)) = 0$.

After dividing equation (4.1) by $\mathrm{F}(n)$, substituting $\mathrm{F}(n+1)/\mathrm{F}(n)$ by its corresponding q-Gosper–Petkovšek representation and clearing common denominators one obtains:

$$\sum_{j=0}^{J} z^j\,\mathrm{C}_j(n)\,\mathrm{P}(n+j) = 0, \tag{4.2}$$

where

$$\mathrm{C}_j(n) := \sum_{k=0}^{\infty} c_{j,k}(q^n)^k := \sigma_j(n) \left(\prod_{i=0}^{j-1} \mathrm{Q}(n+i)\right) \left(\prod_{i=j}^{J-1} \mathrm{R}(n+i)\right) \in \mathbb{F}[q^n]\,.$$

Using Lemma 4.1 it is easy to deduce the two conditions

$$\mathrm{Q}(n) \text{ divides } \sigma_0(n), \qquad \text{and} \qquad \mathrm{R}(n) \text{ divides } \sigma_J(n-J+1),$$

from the above equation. Besides, one can derive

$$\sum_{j=0}^{J} c_{j,\lambda}\,z^j = 0, \qquad \lambda = \min\{\operatorname{ldeg}(\mathrm{C}_j(n)) \mid j = 0, \ldots, J\} \tag{4.3}$$

by equating the coefficients of $(q^n)^\lambda$ in (4.2).

Now the q-Petkovšek algorithm for each possible choice[†] of $\mathrm{Q}(n)$, $\mathrm{R}(n)$ computes possible values of z, i.e. the set

$$\mathcal{Z} = \Big\{x \in \mathbb{F} \;\Big|\; \textstyle\sum_{j=0}^{J} c_{j,\lambda}\,x^j = 0\Big\}.$$

[†]$\mathrm{P}(n)$ is called monic if the leading coefficient is one.

[†]Possible choices for $\mathrm{Q}(n)$, e.g., are all combinations of non-constant monic divisors (w.r.t. n) of $\sigma_0(n)$. Since all divisors can be assumed to be monic, there are only finitely many choices.

Once we know these, we substitute them into equation (4.2) and check whether there is a polynomial solution $\mathrm{P}(n)$.

This means that the algorithm is rather slow if the coefficients $\sigma_0(n)$ and $\sigma_J(n)$ have many factors, as we have to try a lot of combinations. Thus to apply the algorithm, it is really important to have an efficient implementation.

We observed that the number of computations could be substantially decreased by grouping the elements of $\mathcal{Z}$ into equivalence classes according to

$$x \sim \tilde{x} \qquad \Longleftrightarrow \qquad x = q^i\,\tilde{x} \qquad \text{for some } i \in \mathbb{Z}.$$

Assuming that $\mathrm{F}(n)$ is a solution of equation (4.1), $(z, \mathrm{P}(n), \mathrm{Q}(n), \mathrm{R}(n))$ the q-Gosper–Petkovšek representation of $\mathrm{F}(n+1)/\mathrm{F}(n)$, and $z = \tilde{z}\,q^i$ for some integer i, we obtain

$$z\,\frac{\mathrm{P}(k+1)}{\mathrm{P}(k)}\,\frac{\mathrm{Q}(k)}{\mathrm{R}(k)} = \tilde{z}\frac{\tilde{\mathrm{P}}(k+1)}{\tilde{\mathrm{P}}(k)}\,\frac{\mathrm{Q}(k)}{\mathrm{R}(k)}, \qquad \text{with } \tilde{\mathrm{P}}(n) = \left(q^k\right)^i \mathrm{P}(n).$$

This means that instead of searching for polynomial solutions $\mathrm{P}(n)$ of (4.2) for every $z \in \mathcal{Z}$ it suffices to search for all *Laurent* polynomial solutions $\mathrm{P}(n)$ for *one* representative of each equivalence class in $\mathcal{Z}$. By dividing $\mathcal{Z}$ into equivalence classes the computation times usually are halved.

Our implementation `qrecsolve` finds all q-hypergeometric solutions of recurrence equation (3.8) that was stored in the variable `qDixon_RE`, after 146 s:†

```
> Sn:= qrecsolve(qDixon_RE,q,S(n),return=qhypergeometric);
```

$$\mathrm{Sn} := \left[\left[\frac{\mathrm{qpochhammer}\left(q^{c+1+b}, q, n\right)}{\mathrm{qpochhammer}(q, q, n)}, 0 <= n\right]\right].$$

The optional argument `return=qhypergeometric` advises the procedure to return the q-hypergeometric term $\mathrm{S}(n)$ and not the ratio $\mathrm{S}(n+1)/\mathrm{S}(n)$ which is the default. Modulo a constant factor w. r. t. n this should be equivalent to the right-hand side of (3.7):

```
> qsimplify(Sn[1][1]*qpochhammer(q,q,n)*qpochhammer(q,q,b)*
  qpochhammer(q,q,c)/qpochhammer(q,q,n+b+c));
```

$$\frac{(-1+q^b)\,\mathrm{qpochhammer}(q, q, c)}{(-1+q^{b+c})\,\mathrm{qpochhammer}\left(q^b, q, c\right)}.$$

Since $\mathrm{Q}(n)$ and $\mathrm{R}(n)$ are divisors of $\sigma_0(n)$ and $\sigma_J(n-J+1)$, respectively, we can choose to try only possible combinations of factors of the σ's obtained by rational or by a complete factorization. Of course the latter method is eventually slower, but we are sure to obtain all solutions. If we are not interested in solutions over any extension field, however, we might prefer rational factorization. This can be accomplished by using the additional argument `split=false` in `qrecsolve`. If we do this for the q-Dixon example, the solution is already found after 42 s.†

†With Petkovšek's implementation `qHyper` (version from July 1995) and MATHEMATICA 3.0 no result was obtained after one hour. Note that we had to supply the optional arguments `Solutions->All` and `Quadratics->True` to obtain a behavior equivalent to our procedure `qrecsolve`.

† Without `Quadratics->True` Petkovšek's implementation needs 290 s.

4.2. *q*-HYPERGEOMETRIC SOLUTIONS OF INHOMOGENEOUS RECURSIONS

From the computational point of view the case of an inhomogeneous recurrence equation is much easier. If $\mathrm{F}(n)$ is a q-hypergeometric solution of the recurrence equation

$$\sum_{j=0}^{J} \sigma_j(n)\,\mathrm{F}(n+j) = \mathrm{G}(n), \qquad \sigma_0(n), \ldots, \sigma_J(n) \in \mathbb{F}[q^n] \tag{4.4}$$

with given q-hypergeometric term $\mathrm{G}(n)$, then $\mathrm{L}(n) = \mathrm{F}(n)/\mathrm{G}(n)$ is in $\mathbb{F}(q^n)$. To solve equation (4.4) we divide it by $\mathrm{G}(n)$ and search for all *rational* solutions $\mathrm{L}(n)$ of

$$\sum_{j=0}^{J} \left(\sigma_j(n) \prod_{i=0}^{j-1} \gamma(n+i) \right) \mathrm{L}(n+j) = 1, \qquad \gamma(n) = \frac{\mathrm{G}(n+1)}{\mathrm{G}(n)} \in \mathbb{F}(q^n)\,. \tag{4.5}$$

Abramov (1995) showed how to determine a multiple of the denominator of $\mathrm{L}(n)$, therefore leaving the easier problem of finding *polynomial* solutions of inhomogeneous recurrence equations.‡

As an example, we search for all q-hypergeometric solutions $\mathrm{F}(k)$ of the inhomogeneous recurrence equation

$$\begin{aligned} q^k\,(q+1)\,(q-1)\,(q^{k+1}-1)\,\mathrm{F}(k+2) \;+\; (q+1)\,(1-q^{k+1})\,(q^k\,(a\,q-a)+a-1) \\ \times \mathrm{F}(k+1) \;+\; (a+1)\,(a-1)\,(q^{k+1}-1)\,\mathrm{F}(k) = \frac{(q^k+1)\,(q^k-1)\;(a;q)_k}{(q;q)_k} \end{aligned}$$

by the computation

```
> qrecsolve(q^k*(q+1)*(q-1)*(q^(k+1)-1)*F(k+2)+(q+1)*
  (1-q^(k+1))*(q^k*(a*q-a)+a-1)*F(k+1)+(a+1)*(a-1)*(q^(k+1)-1)*F(k)=
  (q^k+1)*(q^k-1)*qpochhammer(a,q,k)/qpochhammer(q,q,k),q,F(k));
```

$$\left[\frac{\left(1-q^k\right) \mathrm{qpochhammer}(a,q,k)}{\mathrm{qpochhammer}(q,q,k)} \right]$$

4.3. POWER SERIES SOLUTIONS

Let $\mathbb{F}_{q\text{-hyp.}}[[q^n]]$ denote the subset of the ring $\mathbb{F}[[q^n]]$ of formal power series over $\mathbb{F}$ whose coefficients form a q-hypergeometric sequence. Abramov *et al.* (1998) showed how to search for solutions $\mathrm{F}(n) \in \mathbb{F}_{q\text{-hyp.}}[[q^n]]$ of

$$\sum_{j=0}^{J} \sigma_j(n)\,\mathrm{F}(n+j) = \sum_{k=0}^{\infty} b_k \left(q^n\right)^k, \tag{4.6}$$

where b_k is a q-hypergeometric term w. r. t. k, based on the following lemma:

‡ To determine polynomial solutions one proceeds as described for the q-Gosper algorithm.

LEMMA 4.2. *Assume* $\mathrm{F}(n) = \sum_{k=0}^{\infty} f_k \,(q^n)^k \in \mathbb{F}_{q\text{-hyp.}}[[q^n]]$ *is a solution of equation* (4.6) *with*

$$\sigma_j(n) = \sum_{i=0}^{\delta} s_{j,i} \big(q^n\big)^i \in \mathbb{F}[q^n]\,, \qquad j = 0, \ldots, J,$$

and $\delta = \max\big\{\, \deg(\sigma_j(n)) \;\big|\; j = 0, \ldots, J \big\}$. *Then the recurrence equation*

$$\sum_{i=0}^{\delta} \left(\sum_{j=0}^{J} s_{j,\delta-i}\, q^{i\,j} \big(q^l\big)^j \right) f_{l+i} = b_{l+\delta}, \tag{4.7}$$

is valid for f_l *with* $l \in \mathbb{N}_0$ *and*

$$\sum_{i=0}^{l} \left(\sum_{j=0}^{J} s_{j,l-i}\, q^{i\,j} \right) f_i = b_l, \qquad l = 0, 1, \ldots, \delta - 1. \tag{4.8}$$

Thus we can determine all solutions $\mathrm{F}(n) \in \mathbb{F}_{q\text{-hyp.}}[[q^n]]$ of equation (4.6) by using Lemma 4.2 and applying q-Petkovšek's algorithm if the right-hand side is zero, and otherwise Abramov's algorithm to determine the coefficients f_l via (4.7). Then we have to check if we can adjust the solution for f_l so that the initial conditions (4.8) are fulfilled.

With the option `solution=series`, the procedure `qrecsolve` searches for formal power series solutions by the above algorithm. For example, given the recurrence equation†

$$q^2\, q^{2\,n}\, \mathrm{F}(n+3) + (1+q)\, q^n\, \mathrm{F}(n+2) + (1-q^n)\, \mathrm{F}(n+1) - \mathrm{F}(n) = 0,$$

we obtain the solution:

```
> qrecsolve(q^2*q^(2*n)*F(n+3)+(1+q)*q^n*F(n+2)+(1-q^n)*F(n+1)-F(n),
  q,F(n),solution=series);
```

$$\left[\sum_{\_i=0}^{\infty} \frac{\_C1\; q^{\_i^2} \big(q^n\big)^{\_i}}{\mathrm{qpochhammer}(q, q, \_i)} \right].$$

Now we want to use this method to introduce an extension of q-Gosper's algorithm, by considering Example (2.9) again. First we compute polynomials $\mathrm{P}(k), \mathrm{Q}(k), \mathrm{R}(k)$ according to (2.3) for $\mathrm{F}_n(k)$ defined by (2.9)

$$\mathrm{P}(k) = 1, \quad \mathrm{Q}(k) = a\, q^k - 1, \qquad \text{and} \qquad \mathrm{R}(k) = a\, q^n \big(q^{k+1} - 1\big).$$

The recurrence equation for $\mathrm{X}(k)$ (see (2.5)) is then

$$\big(a\, q^k - 1\big)\, \mathrm{X}(k) \;-\; a\, q^n \big(q^k - 1\big)\, \mathrm{X}(k-1) = 1.$$

We know already that there is no Laurent polynomial solution $\mathrm{X}(k)$. However, computing a degree bound for $\mathrm{X}(k)$ shows that $\mathrm{X}(k)$ could be a polynomial of degree n. Thus we might try to look for a solution $\mathrm{X}(k) \in \mathbb{F}_{q\text{-hyp.}}\big[\big[q^k\big]\big]$:

```
> qrecsolve((a*q^k-1)*X(k)-a*q^n*(q^k-1)*X(k-1)=1,q,X(k),solution=series);
```

† Abramov *et al.* (1998, Section 5.3, Example 8).

$$\left[\sum_{\_i=0}^{\infty}\left(\frac{\text{qpochhammer}\left(q^{-n},q,\_i\right)\ q^{\_i}\left(q^{k}\right)^{\_i}}{\left(a\,q^{n}-q^{\_i}\right)\text{qpochhammer}\left(\dfrac{q^{-n}}{a},q,\_i\right)}\right)\right].$$

Note that $\left(q^{-n};q\right)_{\_i}=0$ for $\_i>n$, i.e. $\mathrm{X}(k)$ is a polynomial in q^k of degree n as expected. With equation (2.6) we can now build *one* antidifference $\mathrm{G}(k)$ for all $n\in\mathbb{N}_0$. Our procedure `qgosper` can do all the necessary calculations:

```
> assume(n,posint);
> qgosper(qpochhammer(a,q,k)/qpochhammer(q,q,k)/(a*q^n)^k,q,k,series=true);
```

$$\frac{\left(-1+q^{k}\right)a\,q^{n}\,\text{qpochhammer}(a,q,k)}{\left(a\,q^{n}\right)^{k}\,\text{qpochhammer}(q,q,k)}\sum_{\_i=0}^{n}\left(\frac{\text{qpochhammer}\left(q^{-n},q,\_i\right)\ q^{\_i}\left(q^{k}\right)^{\_i}}{\left(a\,q^{n}-q^{\_i}\right)\text{qpochhammer}\left(\dfrac{q^{-n}}{a},q,\_i\right)}\right).$$

Note that after a second `qgosper` delivers an antidifference for *all* $n\in\mathbb{N}$. This should be compared with the timings in Table 1.

4.4. description of the Maple procedure `qrecsolve`

Calling Sequence:

`qrecsolve(RE,q,f(n),...);`

Parameters:

- `RE` — an algebraic expression or an equation, the recurrence equation for f(n)
- `q` — a name
- `f` — a name, the recurrence function
- `n` — a name, the recurrence variable
- `...` — a sequence of optional arguments (of type name=name)

Input/Output Description:

This procedure decides whether a given recurrence equation with polynomial coefficients in q^n has any q-hypergeometric solutions f(n). If no solution exists, an empty list is returned. Otherwise a list is returned, where each solution is stored in another list.

Options and Remarks:

- Using the option `solution` with one of the possible values `polynomial`, `rational`, `qhypergeometric`, or `series` tells `qrecsolve` to search only for this type of solutions. When set to `series`, `qrecsolve` returns all formal power series solutions of the form $\mathrm{f}(n) = \sum_{k=0}^{\infty} a_k \, (q^n)^k$. The default is to search for q-hypergeometric solutions.
- With the option `return` one can control the way solutions are returned. It may be set to `downratio`, `upratio`, `downrec`, `uprec`, or `qhypergeometric`; the default is `uprec`, meaning that an upward recurrence equation of first order is returned for each possible solution $\mathrm{f}(n)$. Using the value `downratio`, `qrecsolve` will return the ratio $\mathrm{f}(n)/\mathrm{f}(n-1)$. With `return=qhypergeometric` the procedure will return a list of lists where each sublist consists of two entries. The first element is a q-hypergeometric solution $\mathrm{f}(n)$ of the recurrence equation and the second element contains restrictions on n.
- The option `split` controls to some extent the introduction of additional roots which are required to solve the recurrence equation. The leading and trailing coefficient of the recurrence equation have to be factored completely in order to obtain all solutions. With `split` set to `false` only rational factorization is used. Thus by specifying `split=false` the procedure might run faster but some solutions—which require the introduction of new roots—may be lost. By default it is set to `true`.

5. Conclusion

Our package `qsum.mpl` is an efficient implementation of the ***q***-Gosper, q-Zeilberger, q-Petkovšek and similar algorithms. It is the first package which combines all these algorithms which are useful tools to deal with problems associated with q-hypergeometric series, in particular with q-analogues of orthogonal polynomials. Particularly, the combination of q-Zeilberger with q-Petkovšek can be used to decide whether a definite q-hypergeometric sum has a representation as a q-hypergeometric term.

Acknowledgements

We would like to thank Axel Riese for explaining some optimizations he did to improve the performance of his procedure `qZeil` that also helped decreasing the computation times of our version.† Furthermore we thank the anonymous referees for their helpful comments and suggestions. One of the referees brought the paper Bauer and Petkovšek (1999) to our attention which also presents a multibasic version of Gosper's algorithm. Of course our implementation can be used to compute the multibasic examples given in their paper.

References

Abramov, S. A. (1995). Rational solutions of linear difference and q-difference equations. *Programm. Comput. Softw.*, **6**, 273–278. (1995). Translated from Programmirovanie, **6**, 3–11.

Abramov, S. A., Bronstein, M., Petkovšek, M. (1995). On polynomial solutions of linear operator equations. In *Proceedings of ISSAC'95, Montreal, Canada*, Levelt, T. ed., pp. 290–296. New York, ACM Press.

† See the second idea in Appendix B.9 of Riese (1997).

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

Askey, R., Wilson, J. A. (1985). Some basic hypergeometric polynomials that generalize Jacobi polynomials. *Mem. Am. Math. Soc.*, **319**, 1–55.

Bauer, A., Petkovšek, M. (1999). Multibasic and mixed hypergeometric Gosper type algorithms. *J. Symb. Comput.*, **28**, 711–735.

Böing, H. (1998). Theorie and algorithmen zur $\boldsymbol{q}$-hypergeometrischen summation, Master's Thesis, Freie Universität, Berlin, Germany.

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

Gasper, G., Rahman, M. (1997). Errata to "Basic Hypergeometric Series". Electronic version available online at `http://www.math.nwu.edu/preprints/gasper/bhserrata`.

Koekoek, R., Swarttouw, R. F. (1994). The Askey-scheme of hypergeometric orthogonal polynomials and its q-analogue. Technical Report 94–05, Technische Universiteit Delft, Faculty of Technical Mathematics and Informatics, Delft. Electronic version available online at `http://www.can.nl/~renes`.

Koepf, W. (1996). Summation in Maple. *Maple Tech. Newslett.*, **3**, 26–32.

Koepf, W. (1998). *Hypergeometric Summation*, Braunschweig/Wiesbaden, Vieweg.

Koornwinder, T. H. (1993). On Zeilberger's algorithm and its q-analogue: a rigorous description. *J. Comput. Appl. Math.*, **48**, 93–111.

Paule, P. (1994). Short and easy computer proofs of the Rogers–Ramanujan identities and of identities of similar type. *Electron. J. Comb.*, **1**.

Paule, P., Riese, A. (1997). A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated approach to q-hypergeometric telescoping. In *Fields Proceedings of the Workshop on 'Special Functions, q-Series and Related Topics', Toronto, Ontario*, Fields Institute for Research in Mathematical Sciences at University College.

Petkovšek, M., Wilf, H. S., Zeilberger, D. (1996). $A = B$. A. K. Peters.

Riese, A. (1995). A Mathematica q-analogue of Zeilberger's algorithm for proving q-hypergeometric identities, Master's Thesis, Research Institute for Symbolic Computation, J. Kepler University, Linz, Austria.

Riese, A. (1996). A generalization of Gosper's algorithm to bibasic hypergeometric summation. *Electron. J. Comb.*, **3**.

Riese, A. (1997). Contributions to symbolic q-Hypergeometric summation, Ph.D. Thesis, Research Institute for Symbolic Computation, J. Kepler University.

Wilf, H. S., Zeilberger, D. (1992). An algorithmic proof theory for hypergeometric (ordinary and 'q') multisum/integral identities. *Invent. Math.*, **108**, 575–633.

Received 15 January 1998
Accepted 23 December 1998