Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

HARALD BÖING AND WOLFRAM KOEPF

Algorithms for q-hypergeometric Summation in Computer Algebra

Preprint SC 98-02 (January 1998)

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*

In this paper we present a short description of q-analogues of Gosper's, Zeilberger's, Petkovšek's and related algorithms. Furthermore we introduce our corresponding MAPLE implementations and show how they can be applied to prove or even derive identities associated with q-series.

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).

Surprisingly enough, 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.* (1997) for Petkovšek.

As a general reference for q-hypergeometric (basic) series we refer to Gasper and Rahman (1990). We sketch the underlying theory only briefly and emphasize on the application of our MAPLE package `qsum.mpl` which is an implementation of those algorithms, containing an m-dimensional version of the q-Gosper algorithm, and some other generalizations. This package and an extensive help database for MAPLE V Release 4 is available by e-mail request from `koepf@imn.htwk-leipzig.de`. All examples shown in this paper were computed with MAPLE V Release 4 on a Sun Sparc 20.

For MATHEMATICA the package `qZeil.m` written by Riese is an excellent implementation of q-Gosper's and q-Zeilberger's algorithm.[1]

Throughout this paper by $\mathbf{q}$ the vector $(q_1, q_2, \ldots, q_m)$, $m \in \mathbb{N}$ is denoted, and $\mathbb{F}$ is an abbreviation for $\mathbb{K}(q_1, q_2, \ldots, q_m)$, $\mathbb{K}$ denoting the field $\mathbb{Q}$ extended by some parameters.[2]

[1] The package `qZeil` is available by e-mail request to `Peter.Paule@risc.uni-linz.ac.at`.

[2] 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.

2. q-Gosper

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

$$\frac{\mathrm{F}(k+1)}{\mathrm{F}(k)} \in \mathbb{F}\big(q_1^k, \ldots, q_m^k\big)$$

a **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*[3] $\mathrm{G}(k)$ for $\mathrm{F}(k)$ makes the evaluation of *definite sums* over $\mathrm{F}(k)$ easy:

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

The **q**-Gosper algorithm is based on the following two observations:

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

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

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 details:

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

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

and

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

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 $\mathrm{X}(k) \in \mathbb{F}\big[q_1^k, q_1^{-k}, \ldots, q_m^k, q_m^{-k}\big]$ 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.

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

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

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

After loading our package in MAPLE via `read ‘qzeil.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);
```

$$\mathit{result} := \frac{\mathrm{qpochhammer}(c,q,k)\ \mathrm{qpochhammer}(a,p,k)\ (-1+q^k)\,(-c+p^k\,a)\,c^{(-k)}}{\mathrm{qpochhammer}\left(\dfrac{a\,p}{c},p,k\right)\ \mathrm{qpochhammer}(q,q,k)\ (-1+a)\,(c-1)}$$

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 **q**-hypergeometric terms. It was designed to simplify ratios of **q**-hypergeometric terms like $\mathrm{F}(k+1)/\mathrm{F}(k)$ to decide rationality. In other cases the procedure might fail to give the simplest result. Then one can try to use the procedure `qsimplify` that does a more rigid simplification.

Generally the most time consuming part in **q**-Gosper's algorithm is its second step, which is usually done in two parts:[4]

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 yet unknown coefficients c_i in Equation (2.5) and solve the resulting *linear* system by comparing the coefficients of $(\mathbf{q}^k)^i$.

This method involves solving a system of $h-l+1$ linear equations which is inefficient if the difference $h-l$ is large. In 1995 Abramov, Bronstein and Petkovšek 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.* suggest to use their algorithm if the difference $h-l$ is greater or equal to the order of the recurrence equation that is to be solved.

[4] To simplify the notation, we just describe the case $m=1$.

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

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

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:

```
> 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 nonpositive integer $n \in \mathbb{N}_0$ there exists an antidifference, where the resulting recurrence relation for $\mathrm{X}(k)$, i. e.

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

implies $\mathrm{X}(k)$ being a polynomial of degree n. Thus for $n=1$ we get e. g.[5]

```
> 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`[6] 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.* described above that we implemented for $m=1$.
`solve`: This is MAPLE's builtin `solve` procedure.
`gausselim`: An implementation of Gaussian implementation, 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 (1995).
`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.

The following table 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:[7]

[5] By 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`.
[6] This also applies to `qsumrecursion`, introduced in the next section.
[7] The occurring dashes (—) in the table mean that no result was obtained within two hours.

Table 1. Timings for different solve methods

n	10	20	30	40	50	60	70	80	90	100
`solve`	6 s	106 s	1426 s	—	—	—	—	—	—	—
`gausselim`	2 s	8 s	27 s	63 s	146 s	270 s	466 s	770 s	1188 s	1642 s
`ABP`	1 s	2 s	6 s	21 s	46 s	96 s	160 s	252 s	378 s	519 s

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

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

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's 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));
```

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

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

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

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

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* $\mathrm{S}(n)$

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

where $\mathrm{F}(n,k)$ is a q-hypergeometric term w.r.t. n and k. It uses the q-Gosper algorithm to determine an antidifference $\mathrm{G}(n,k)$ 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[8]

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

Note that 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 $\sigma_0(n),\ldots,\sigma_J(n)$. Thus the only change required is to add the unknowns $\sigma_0(n),\ldots,\sigma_J(n)$ to the variables of the linear system that has to be solved.

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

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

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 knows $\mathrm{G}(n,k)$:

1 Write down the corresponding recurrence equation (3.3) for the summand $\mathrm{F}(n,k)$ and divide it by $\mathrm{F}(k)$. 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*.[9]

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 *compact 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).

Assume we want to prove the q-Pfaff-Saalschütz identity

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

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

[9] after simplifying the ratios of the q-hypergeometric terms

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

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

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

```
> 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[10] 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 gets, 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];
```

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

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

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

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. But be careful, to avoid interference with the global variable k we defined k to be local.[11] 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`:

[10] As we expected it turned out to be zero.
[11] I. e., a substitution, such as `subs(k=k+1,term)` would *not* change anything!

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

$$0$$

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 *compact support*, i.e.

$$\frac{(a;q)_k\,(b;q)_k\,(q^{-n};q)_k\,q^k}{(c;q)_k\,(a\,b\,q^{1-n}/c;q)_k\,(q;q)_k}=0, \qquad \text{for } k\in\left\{j\in\mathbb{Z}\;\middle|\;j\le -1 \text{ or } j\ge n+1\right\}$$

the easiest way to prove `_qsumrecursion_proof[_recursion]` now, is by summing over k from 0 to ∞, showing that the inhomogeneous part is zero.

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

$$\sum_{k=-n}^{n}(-1)^k\,q^{k\,(3\,k-1)/2}\begin{bmatrix}n+b\\n+k\end{bmatrix}_q\begin{bmatrix}n+c\\c+k\end{bmatrix}_q\begin{bmatrix}b+c\\b+k\end{bmatrix}_q \;=\; \frac{(q;q)_{n+b+c}}{(q;q)_n\,(q;q)_b\,(q;q)_c}, \tag{3.6}$$

where the q-binomial coefficient is defined by

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

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

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

$$\begin{aligned}
&\mathit{qDixon_RE} := \left(-q^{2\,n}+q\right)\left(q^n+1\right)\left(q^n-1\right)q^3\,\mathrm{S}(n) \;+\; q\,\big(q^5+q^4-q^{4+n+c+b}-\\
&q^{2\,n+3}-q^{n+3+c}-q^{n+3+c+b}+q^3+q^{2\,n+3+c}+q^{2\,n+3+b}-q^{n+3+b}-q^{3\,n+2}+\\
&q^{2\,n+2+b}+q^{4\,n+2+c+b}+q^{2\,n+2+c}-q^{3\,n+2+b}-q^{3\,n+2+c}-q^{2\,n+2+c+b}-q^{1+3\,n}+\\
&q^{4\,n+1+c+b}+q^{4\,n+c+b}\big)\,\mathrm{S}(n-1) \;-\; \left(q-q^{n+c+b}\right)\big(-q^{3\,n+c+b+1}+q^5+q^{2\,n+2}-\\
&q^{3\,n+c+b}+q^4-q^{3\,n+2+c+b}+q^6-q^{4+n+c+b}+q^{2\,n+3+b}-q^{n+3+c}+q^{2\,n+3+c}-\\
&q^{n+3+b}-q^{4+n+c}+q^{2\,n+2+b}+q^{2\,n+2+c}-q^{4+n+b}\big)\,\mathrm{S}(n-2) \;+\\
&\left(q^2-q^{n+c+b}\right)\left(q-q^{n+c+b}\right)\left(q^2-q^{n+b}\right)\left(q^2-q^{n+c}\right)\mathrm{S}(n-3) \;=\; 0,
\end{aligned} \tag{3.7}$$

which is definitely not the result we want. Here we have an example (of quite a few), where the q-Zeilberger algorithm doesn't find a recurrence equation of *minimal* order, which was pointed out by Paule and Riese (1995). Paule introduced 1994 the method of *creative symmetrizing*—based on the following lemma—which resolves 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*

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

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:[12]

```
> M:= qsimpcomb(subs(k=-k,term)/term,assume=[k,integer]);
```

$$M := q^k$$

If we now apply `qsumrecursion` to the symmetrized summand we get:

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

$$\big(q^n-1\big)\,\mathrm{S}(n) \;-\; \big(-1+q^{n+b+c}\big)\,\mathrm{S}(n-1) \;=\; 0$$

The following table is an extract of the one given in Paule and Riese (1995) with the most time consuming examples, where we used the newest version of their package `qZeil` (Version 1.8) with Mathematica 3.0:

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

equation[13]		order	`qsumrecursion`	Turbo-`qZeil`
(II.18)	lhs	2	7 s	7 s
	rhs	2	4 s	7 s
(III.25)	lhs	3	10 s	18 s
	rhs	3	26 s	43 s
(III.28)	lhs	2	34 s	51 s
	rhs	2	36 s	64 s

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

[12] To remove factors like $((-1)^k)^2$ the information that k is an integer is passed to our simplification procedure `qsimpcomb`.

[13] All equations numbers refer to the appendix of the book of Gasper and Rahman (1990).

The q-Zeilberger algorithm gives also a very simple method to deduce recurrence equations for orthogonal polynomials from the Askey-Wilson scheme (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));
```

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

and

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

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

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

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

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

$$Q_n(x;a,b|q) \;=\; \frac{(a\,b;q)_n}{a^n}\,{}_3\phi_2\left(\begin{matrix} q^{-n}, a\,e^{i\,\theta}, a\,e^{-i\,\theta} \\ a\,b, 0 \end{matrix} \,\middle|\, q;\, q\right) \tag{3.8}$$

$$\;=\; \left(b\,e^{-i\,\theta};q\right)_n\,e^{i\,n\,\theta}\,{}_2\phi_1\left(\begin{matrix} q^{-n}, a\,e^{i\,\theta} \\ b^{-1}\,q^{1-n}\,e^{i\,\theta} \end{matrix} \,\middle|\, q;\, b^{-1}\,q\,e^{-i\,\theta}\right). \tag{3.9}$$

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));
```

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

```
> 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));
```

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

Since the two recurrence equations agree, (3.8) and (3.9) 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 confirmed.

4. q-Petkovšek

In Abramov *et al.* (1997) the authors give a few algorithms to find solutions of special types of homogeneous and inhomogeneous recurrence equations with polynomial coefficients. Petkovšek implemented the procedure `qHyper`[14] in MATHEMATICA that finds q-hypergeometric solutions of homogeneous recurrence equations.

4.1. q-HYPERGEOMETRIC SOLUTIONS OF HOMOGENEOUS RECURSIONS

Assume we want to determine all q-hypergeometric 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* $\mathrm{L}(n) \in \mathbb{F}(q^n)$ *there exists a 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) $\quad \mathrm{L}(n) = z\,\dfrac{\mathrm{P}(n+1)}{\mathrm{P}(n)}\,\dfrac{\mathrm{Q}(n)}{\mathrm{R}(n)},$

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

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

(iv) $\quad \mathrm{P}(n),\mathrm{Q}(n),$ *and* $\mathrm{R}(n)$ *are monic*[15], *and* $\mathrm{ldeg}(\mathrm{P}(n)) = 0$.

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 canceling common denominators one gets:

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

where

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

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

[14] The corresponding package `qHyper.m` (and the hypergeometric version `Hyper.m`) can be obtained at `http://www.mat.uni-lj.si/dwnld.htm`.
[15] $\mathrm{P}(n)$ is called monic if the leading coefficient is one.

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

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

Now the q-Petkovšek algorithm for each possible choice[16] of $\mathrm{Q}(n)$, $\mathrm{R}(n)$ computes possible values of z, i. e. the set

$$\mathcal{Z} \;=\; \left\{x \in \mathbb{F} \;\middle|\; \sum\nolimits_{j=0}^{J} c_{j,\lambda}\, x^j = 0\right\}.$$

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

This means 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

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

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

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

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 $\mathcal{Z}$.

Our implementation `qrecsolve` finds all q-hypergeometric solutions of recurrence equation (3.7) that was stored in the variable `qDixon_rec`, after 290 s:[17]

```
> Sn:= qrecsolve(qDixon_rec,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 (3.6):

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

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

[16] 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 are monic, there are only finitely many choices.

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

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 get 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 105 s.[18]

By dividing $\mathcal{Z}$ into equivalence classes the computation times usually are halved: For `qDixon_rec`, e. g., without this improvement our implementations needs 589 s and 234 s with `split` set to `true` and `false`, respectively.

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

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

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.[19]

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

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

by the computation

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

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

[18] Without `Quadratics->True` Petkovšek's implementation needs 1244 s.
[19] To determine polynomial solutions one proceeds as described for the q-Gosper algorithm.

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. In Abramov *et al.* (1997) the authors show how to search for solutions $\mathrm{F}(k) \in \mathbb{F}_{q\text{-hyp.}}[[q^n]]$ 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:

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

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

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

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

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. E.g., given the recurrence equation[20]

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

we get 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);
```

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

[20] (Abramov *et al.*, 1997), Section 5.3, Example 8.

Now we want to use this method to introduce an extension of q-Gosper's algorithm, by considering Example (2.9). 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)

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

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

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

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 $\mathrm{X}(k) \in \mathbb{F}_{q\text{-hyp.}}[[q^n]]$:

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

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

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);
```

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

Note that after 2 s `qgosper` delivers an antidifference for *all* $n \in \mathbb{N}$. This should be compared with the timings in Table 1. The declaration `assume(n,posint);` helped the procedure `qgosper` to determine that the sum is not infinite since the summand is zero for $\_i > n$.

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 sums. Especially can the combination of q-Zeilberger with q-Petkovšek be used to determine whether a definite q-hypergeometric sum has a representation as a q-hypergeometric term.

Acknowledgment

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.[21]

[21] See the second idea in Appendix B.9 of Riese (1997)

References

Abramov, S. A. (1995). Rational solutions of linear difference and q-difference equations. *Programming and Comput. Software*, **6**:273–278. Transl. from Programmirovanie (1995) 6, 3-11.

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

Abramov, S. A., Paule, P., Petkovšek, M. (1997). q-hypergeometric solutions of q-difference equations. *Discrete Math.*, to appear.

Böing, H. (1998). Theorie und Algorithmen zur **q**-hypergeometrischen Summation. Master's thesis, Freie Universität Berlin, Germany.

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

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 at `http://www.can.nl/~renes`.

Koepf, W. (1996). Summation in Maple. *Maple Technical Newsletter*, **3**(2):26–32.

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

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

Paule, P. (1994). Short and easy computer proofs of the Rogers-Ramanujan Identities and of identities of similar type. *The Electronic Journal of Combinatorics*, **1**:1–9.

Paule, P., Riese, A. (1995). 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. *Electr. Journal of Combinatorics*, **3**:1996.

Riese, A. (1997). *Contributions to Symbolic* **q**-*Hypergeometric Summation*. PhD thesis, Research Institute for Symbolic Computation, J. Kepler University, Linz, Austria.

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