HANDLING HYPERGEOMETRIC SERIES IN MAPLE

Tom H. Koornwinder
CWI
P.O. Box 4079
1009 AB Amsterdam, The Netherlands

Abstract
This paper discusses and evaluates the standard procedures available in Maple 4.3 for handling hypergeometric series. Some suggestions for improvement are given. The paper concludes with a survey of the algorithms of Gosper and of Zeilberger for possibly evaluating a terminating series of hypergeometric type.

This paper appeared in *Orthogonal polynomials and their applications*, C. Brezinski, L. Gori and A. Ronveaux (eds.), IMACS Annals on Computing and Applied Mathematics 9, Baltzer, 1991, pp. 73–80.

The text of the present version (October 19, 2006) is unchanged except that references have been updated.

1. INTRODUCTION

This paper is an account of some experiences with Maple's performance regarding hypergeometric series. It can be regarded as a consumer's report. The author has tested the library files in Maple 4.3 dealing with hypergeometric series, but he has not yet attempted to write new procedures handling such series. However, some summation procedures developed by Zeilberger [14] and not included in Maple's library will also be described.

Simplification by explicit summation of a series is the main theme of this paper. Such simplification can be realized by look-up from some built-in database or by algorithm. Both approaches will be discussed.

It is hoped that this paper will stimulate both the developers at Waterloo and users writing their own Maple routines in order to improve Maple's capabilities in handling hypergeometric series.

Here are the contents of the paper. Section 2 contains preliminaries on hypergeometric series. Section 3 discusses some typical Maple sessions dealing with summation of hypergeometric series. Section 4 gives an overview of the hypergeometric routines available in Maple. Section 5 contains some suggestions for improvement. Section 6 discusses the merits of the approach by database versus the algorithmic approach. The more technical sections 7 and 8 describe the algorithms of Gosper and Zeilberger. Finally, section 9 offers some further perspectives.

Acknowledgements. The testing was done with Maple version 4.3 implemented at the Sun 4 computer of the Foundation CAN (Computer Algebra Nederland) at the CWI. André Heck was very helpful whenever I had questions on Maple. George Gasper focused my attention on Gosper's algorithm and sent me many test problems. Doron Zeilberger kindly supplied me with the preprints describing his algorithm and emailed me his Maple procedures.

2. PRELIMINARIES ON HYPERGEOMETRIC SERIES

A conceptual definition of *hypergeometric series* is a series $\sum_{k=0}^{\infty} c_k$ with $c_0 = 1$ and c_{k+1}/c_k being a rational function of k, say

$$\frac{c_{k+1}}{c_k} = \frac{(a_1+k)\ldots(a_p+k)\,z}{(b_1+k)\ldots(b_q+k)\,(1+k)}\,, \tag{2.1}$$

where $a_1,\ldots,a_k$, $b_1,\ldots b_q$ and z are complex numbers. With the *shifted factorial* notation

$$(a)_k := a(a+1)\ldots(a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)}, \qquad k \in \mathbf{Z}_+, \tag{2.2}$$

the series becomes

$$\sum_{k=0}^{\infty} \frac{(a_1)_k \ldots (a_p)_k\, z^k}{(b_1)_k \ldots (b_q)_k\, k!}\,,$$

which we write compactly as

$${}_pF_q\left[\begin{matrix} a_1,\ldots,a_p \\ b_1,\ldots,b_q \end{matrix}; z\right] \quad \text{or} \quad {}_pF_q(a_1,\ldots,a_p;b_1,\ldots,b_q;z),$$

or, in Maple, as

```
hypergeom([a1, a2, ..., ap], [b1, b2, ..., bq], z)
```

In general, it is required that $b_1,\ldots,b_q \notin 0,-1,-2,\ldots$, since otherwise the denominators in the series will become eventually zero. If, for some i, $a_i = -n$ ($n \in \mathbf{Z}_+$) then all terms with $k > n$ will vanish, so the series will terminate. In the non-terminating case, the ratio test yields the radius of convergence: ∞ if $p < q+1$, 1 if $p = q+1$ and 0 if $p > q+1$. Moreover, if $p = q+1$ then there will be absolute convergence for $|z| = 1$ if

$$\mathrm{Re}\,(a_1 + \cdots + a_p - b_1 - \cdots - b_q) < 0.$$

Note that the hypergeometric series is symmetric both in its upper parameters $a_1,\ldots,a_p$ and its lower parameters $b_1,\ldots,b_q$ and that an equal upper and lower parameter cancel each other, yielding a ${}_{p-1}F_{q-1}$ series. A series without $k!$ in the denominator can be obtained by inserting an upper parameter 1.

Some elementary cases of hypergeometric series are the *exponential series*

$$ {}_0F_0(-;-;z) = e^z \tag{2.3}$$

and the *binomial series*

$$ {}_1F_0(a;-;z) = (1-z)^{-a}. \tag{2.4}$$

The ${}_2F_1$ series is the classical Gaussian series, with Jacobi polynomials (including Gegenbauer, Legendre and Chebyshev polynomials) as terminating cases. Bessel functions can be expressed by means of ${}_0F_1$ series.

We will mostly deal with the $p = q+1$ case, where we call

$$ {}_{r+1}F_r(a, a_1, \ldots, a_r; b_1, \ldots, b_r; z), $$

k-balanced (or just *balanced* for k=1) if

$$ a + a_1 + \cdots a_r + k = b_1 + \cdots + b_r\,, $$

well-poised if

$$ 1 + a = a_1 + b_1 = \cdots a_r + b_r $$

and *very well-poised* if it is well-poised and moreover $a_1 = 1 + a/2$.

There are a number of cases where a ${}_{r+1}F_r$ series of argument 1 or -1 can be evaluated in closed form as a quotient of products of gamma functions. See Bailey [4] and Slater [12]. We list some typical cases:

$$ {}_2F_1(a, b; c; 1) = \frac{\Gamma(c)\,\Gamma(c-a-b)}{\Gamma(c-a)\,\Gamma(c-b)} $$

(*Gauss' summation formula*),

$$ {}_2F_1(a, b; 1+a-b; -1) = \frac{\Gamma(1+a-b)\,\Gamma(1+a/2)}{\Gamma(1+a)\,\Gamma(1+a/2-b)} $$

(*Kummer's summation formula*),

$$ {}_3F_2(a, b, -n; c, 1+a+b-c-n; 1) = \frac{(c-a)_n\,(c-b)_n}{(c)_n\,(c-a-b)_n} \tag{2.5}$$

(*Saalschütz's summation* of terminating balanced ${}_3F_2(1)$),

$$ {}_3F_2(a, b, c; 1+a-b, 1+a-c; 1) = \frac{\Gamma(1+a/2)\,\Gamma(1+a-b)\,\Gamma(1+a-c)\,\Gamma(1+a/2-b-c)}{\Gamma(1+a)\,\Gamma(1+a/2-b)\,\Gamma(1+a/2-c)\,\Gamma(1+a-b-c)} $$

(*Dixon's summation formula* of well poised ${}_3F_2(1)$),

$$ {}_7F_6\left[\begin{matrix} a, & 1+a/2, & b, & c, & d, \\ & a/2, & 1+a-b, & 1+a-c, & 1+a-d, \\ 1+2a-b-c-d+n, & & -n & & \\ -a+b+c+d-n, & & 1+a+n & & \end{matrix}; 1\right] = \frac{(1+a)_n(1+a-b-c)_n(1+a-b-d)_n(1+a-c-d)_n}{(1+a-b)_n(1+a-c)_n(1+a-d)_n(1+a-b-c-d)_n} $$

(*Dougall's summation* of very well-poised 2-balanced terminating ${}_7F_6(1)$).

Note that Gauss' formula is a limit of Saalschütz's formula for $n \to \infty$, while Kummer's formula is a limit of Dixon's formula for $c \to \infty$. Kummer's formula in its turn can be obtained from Dougall's formula by putting $d = a/2$ and next letting $n \to \infty$. It is essential in Saalschütz's and Dougall's identities that n is a nonnegative integer.

We will in particular use Saalschütz's summation formula as an example. It can be more explicitly written as

$$\sum_{k=0}^{n} \frac{(a)_k\,(b)_k\,(-n)_k}{(c)_k\,(1+a+b-c-n)_k\,k!} = \frac{(c-a)_n\,(c-b)_n}{(c)_n\,(c-a-b)_n}\,. \tag{2.6}$$

In particular, for $c = a+b+1$ this becomes

$$\sum_{k=0}^{n} \frac{(a)_k\,(b)_k}{(a+b+1)_k\,k!} = \frac{(a+1)_n\,(b+1)_n}{(a+b+1)_n\,n!}\,. \tag{2.7}$$

Note that in this last sum the summand is independent of the summation boundary n, so it is in fact an *indefinite summation formula*.

Hypergeometric sums are often met in the form of (combinatorial) sums with binomial coefficients, cf. Riordan [11]. For instance, Saalschütz's summation can be equivalently written as

$$\sum_{k=0}^{n} \binom{a+k-1}{k} \binom{c-a-b+n-k-1}{c-a-b-1} \Big/ \binom{c+k-1}{c-b} = \frac{c-b}{b} \binom{c-a+n-1}{n} \Big/ \binom{c+n-1}{b} \tag{2.8}$$

and the specialization $c = a+b+1$ yields

$$\sum_{k=0}^{n} \binom{a+k-1}{k} \Big/ \binom{a+b+k}{a+1} = \frac{a+1}{b} \binom{b+n}{n} \Big/ \binom{a+b+n}{b}. \tag{2.9}$$

Evidently, one hypergeometric sum may have many representations as a sum with binomial coefficients. It is very efficient to try to rewrite a sum $\sum_{k=0}^{n} c_k$ (n may be ∞) as a hypergeometric series by computing the quotient (2.1), since then a systematic search in a table of formulas or a computer database is possible in order to see if an explicit evaluation is already known, cf. Askey [2]. Maple has a procedure `` `convert/hypergeom` `` which performs this conversion of a sum into hypergeometric form, if possible, and this procedure is sometimes called by the general `sum` procedure.

3. SOME SIMPLE MAPLE SESSIONS

In the following partial accounts of Maple sessions we have put `printlevel:=4` in order to get useful information about procedures called by Maple, but afterwards we have suppressed some information which is irrelevant here. Let us first see how Maple deals with the combinatorialist's way (2.8) of entering Saalschütz's sum.

```
> # Saalschutz sum with binomial coefficients

> sum(binomial(a+k-1,k)*binomial(c-a-b+n-k-1,
> c-a-b-1)/binomial(c+k-1,c-b), k=0..n):

sum/indef: indefinite summation

sum/gosper: applying Gosper's algorithm

sum/gosper: Gosper's algorithm fails

sum/def2: definite sum using hypergeometric fcns

convert/hypergeom: hypergeom fcn is
                                                        [-n,a,b] [c,a+1+b-n-c] 1

sum/def2: using hypergeometric summation thms

simplify:hypergeom: trying 3,2 simplifications

simplify: applying  GAMMA  function to expression

> convert(",GAMMA);

                  GAMMA(c-b+1)GAMMA(b)GAMMA(c-a+n)GAMMA(c-b+n)
                  -------------------------------------------
                    GAMMA(n+1)GAMMA(c-a)GAMMA(c-b)GAMMA(c+n)
>
```

When we give this same sum to Maple, but with $c = a + b + 1$ (cf. (2.9)) then it turns out that Gosper's algorithm (to be explained in §7) succeeds:

```
> # indefinite Saalschutz summation with binomial
> # coefficients

> sum(binomial(a+k-1,k)/binomial(a+b+k,a+1),
> k=0..n):

sum/indef: indefinite summation

sum/gosper: applying Gosper's algorithm

sum/gosper: Gosper's algorithm succeeds

> convert(",GAMMA);

                  (n+1)GAMMA(a+n+2)(a+b+n+1)(a+1)GAMMA(b+n+2)
                  ------------------------------------------
                    GAMMA(n+2)GAMMA(a+b+n+2)(b+n+1)(a+n+1)b
>
```

Finally we enter Saalschütz's sum in hypergeometric form (2.5):

```
> # Saalschutz summation in hypergeometric form
> f:=hypergeom([a,b,-n],[c,1+a+b-c-n],1):
> simplify(f);
simplify: applying hypergeom fct to expression
simplify:hypergeom: trying 3,2 simplifications
                hypergeom([a, b, -n], [1 + a + b - c - n, c], 1)
> # Has to specify that n is nonnegative integer
> readlib(sum):
> `simplify/hypergeom`(f,{n});
simplify:hypergeom: trying 3,2 simplifications
simplify: applying  GAMMA  function to expression
                    GAMMA(1+b-c)GAMMA(1+a-c)GAMMA(1+a+b-c-n)
                   GAMMA(1-c-n)/(GAMMA(1+b-c-n)GAMMA(1+a-c-n)
                           GAMMA(1+a+b-c)GAMMA(1-c))
>
```

The definite summation case of Saalschütz's sum is solved, both in the first and in the third example by looking up the identity in a kind of database, which for ${}_3F_2$ type summations is encoded in `simplify/hypergeom/args32`. With the binomial notation the expression was entered as `sum` and then the procedure `sum`, calling various subroutines, eventually arrives at conversion into hypergeometric form, the determination of the type of hypergeometric series and calling of `args32`. Maple assumes that the `n` in `sum(expression, k=0..n)` is a nonnegative integer and passes this information to `args32`. On the other hand, when the expression is entered as `hypergeom` then little or nothing is done with it and `simplify` has to be called in order to simplify the expression. When `simplify` finds `hypergeom` in an expression then it will eventually call `` `simplify/hypergeom` ``, and so on. However, now the information that `n` is nonnegative integer has to be given explicitly in a rather clumsy way by

`` `simplify/hypergeom`(hypergeometric_expression, {n1, n2, ..., nj}) ``

where {n1, n2, ..., nj} is a set of expressions assumed to be nonnegative integers. First, if `sum` was not used earlier in the session, one has to do `readlib(sum)`, since `` `simplify/hypergeom` `` will call `` `sum/nonpostest` ``, which is contained in `sum`.

The indefinite summation case of Saalschütz's identity (second example) can only be entered as a `sum`. (Indeed, substitution of `c:=a+b+1` in `hypergeom([a,b,-n],[c,1+a+b-c-n],1)` would yield `hypergeom([a,b,-n],[a+b+1,-n],1)` and this will be transformed by Maple into the expression `hypergeom([a,b],[a+b+1],1)`, which has a completely different meaning and represents an infinite summation.) Gosper's algorithm will succeed now. So the answer is obtained in a completely different way from the first and third example: by algorithm.

These simple examples already show which services the user can expect from Maple's standard packages when he wants to find explicit answers to summations of hypergeometric type. A first category of users, without much knowledge of special functions and without the habit of writing their own Maple routines, may consider Maple as a black box into which one may enter summation

expressions with summands of elementary type, as in the first two examples. They will be reasonably served by Maple with output giving an explicit evaluation in many but not all theoretically possible cases. A second smaller category of users will have knowledge of special functions, would like to enter expressions in hypergeometric notation and would like to have some modest interaction with Maple in order to guide Maple to the desired output: some transformed or simplified expression. Such users will meet a rather unfriendly interface and many lacking routines, altogether something which is miles away from a special functions laboratory ready for use and almost capable to replace your (old-fashioned) scratch-pad. There is a third minuscule category of users who are familiar with special functions, but are moreover willing, capable and possessing sufficient time to write their own Maple routines. They may arrive at wonderful things, cf. for instance Zeilberger's work [14] to be discussed in section 8.

4. AN OVERVIEW OF HYPERGEOMETRIC ROUTINES IN MAPLE

Maple has a relatively small kernel and a hierarchically ordered library of files containing procedures. These library files can be called by the kernel or by each other or by the user with the `readlib` command. Here follows a (probably not exhaustive) list of files containing hypergeometric routines.

```
hypergeom
simplify
  ./hypergeom
    ./args21
    ./args32
    ./args43
    ./args54
    ./args65
    ./args76
    ./contig21
    ./genred
diff
  ./hypergeom
convert
  ./hypergeom
evalf
  ./hypergeom
series
  ./hypergeom
sum
  ./indef (calls sum/gosper)
  ./gosper
  ./def2 (calls convert/hypergeom)
```

The role of `convert/hypergeom` and of `sum` and its subroutines will be clear from the examples of section 3. I briefly discuss the contents of the other files:

`hypergeom` contains some simple tests about the right format of the arguments of the `hypergeom` function and, in case of numeric argument and parameters, about convergence. Equal upper and lower parameters are cancelled against each other, some (rather arbitrarily chosen) simplifications are already performed and some rewriting in terms of other special functions is done (for the

elementary exponential and binomial series, cf. (2.3), (2.4), but also for the ${}_0F_1$ series in terms of a Bessel function). This library file is called by most other routines dealing with hypergeometric series and remains available afterwards. It can also be `readlib`ed by the user. But otherwise, when the user enters an expression containing `hypergeom`, this will be returned unchanged. If `hypergeom` is called by, say, `simplify` and if it returns, say, a Bessel function then `simplify` will call another subroutine dealing with Bessel functions. We have not listed these subroutines in our survey.
Discussion. It is unsatisfactory that Maple's reaction to expressions containing `hypergeom` depends on preceding events in the session. I think also that Maple should not do any simplifications or conversions without explicit request by the user. For instance, for many purposes it is desirable to have the Bessel function as a hypergeometric series. Then it is unfortunate that Maple will immediately convert this hypergeometric expression back into a Bessel expression.

`simplify/hypergeom/args21` contains formulas for simplification of Gaussian hypergeometric series, such as Gauss' and Kummer's summation formula, but also expressions in terms of elementary functions for special parameter values, cf. the formulas discussed below under `contig` and several others like

$$ {}_2F_1(a/2,-a/2;1/2;-z) = \tfrac{1}{2}\,(\sqrt{1+z}-\sqrt{z})^a + \tfrac{1}{2}\,(\sqrt{1+z}-\sqrt{z})^{-a}. \quad (4.1) $$

The formulas are taken from [1, Ch.15].
Discussion. For $z := \sinh^2 t$ the right hand side of (4.1) simplifies to $\cosh(at)$. However, Maple will not succeed to arrive at this answer, but leaves the user with a complicated expression. It would be advisable to add this simple answer to `args21`, as long as `simplify/trig` is not made more powerful in handling such cases. The same remark applies to several other formulas implemented in `args21` and also to the substitution $z := -\sin^2 t$. By the way, in experimenting with `simplify/trig` I met a curious bug. When `simplify` is applied to a `hypergeom` with trig functions in its argument then `simplify` will call both `simplify/hypergeom` and `simplify/trig`, but the order in which these routines will be applied will vary in an uncontrolled way. If `./trig` is applied first then it will be applied to all parts of the expression, not only the part containing trig functions. Then `./trig`, for a mysterious reason, will transform the list `[a/2,-a/2]` into the quotient `[a,-a]/[2,2]` of two lists. Of course, `./hypergeom` next will not accept this as one of its arguments.

`simplify/hypergeom/args32,...,args76` contain summation formulas for ${}_3F_2,\ldots,{}_7F_6$ of argument 1 and -1, such as Saalschütz's, Dixon's and Dougall's summation formulas. The formulas are taken from Appendix III in [12].
Discussion. It cannot be expected that all explicit summations available in literature are covered by just implementing the formulas from Slater's [12] Appendix. Still there are some omissions which might easily be repaired. For instance, the left hand side of Dougall's summation formula with $d = a/2$ will simplify to a ${}_5F_4$, which is missed by `args54`. Another comment is that many summation formulas follow from specialization of parameters in transformation formulas between two ${}_pF_q$'s. If one of the sides becomes explicitly summable then so will be the other side. For instance, Bailey [4, 4.4(2)] expresses a well-poised ${}_6F_5(-1)$ in terms of a ${}_3F_2(1)$. Then the various summation formulas for ${}_3F_2$'s yield summation formulas for ${}_6F_5$, not all of which are covered by `args65`.

`simplify/hypergeom/contig21` tries to write a ${}_2F_1$ function in terms of more elementary functions by use of iterated contiguous relations (cf. Erdélyi [6, 2.8 (28),(38)]). The iteration will end satisfactorily if it arrives at one of the following cases (probably this list is exhaustive):

$$ {}_2F_1(1,1;2;z) = -z^{-1}\,\log(1-z), $$

$$
{}_2F_1(1/2,1/2;3/2;z^2) = z^{-1}\ \arcsin(z),
$$
$$
{}_2F_1(1/2,1;3/2;-z^2) = z^{-1}\ \arctan(z).
$$

Discussion. The appreciation of this routine depends on one's way of measuring simplification. The result may contain many (although elementary) terms. There are also cases, like `hypergeom([3,1/2],[9/2],z)`, where the routine will return a sum of some elementary terms and some hypergeoms. In such instances the routine could better return the original expression.

`simplify/hypergeom/genred` reduces a ${}_pF_q$ to a hypergeometric series with fewer upper and lower parameters if the difference between an upper and a lower parameter is a positive integer. The relevant formula is

$$
{}_pF_q\left[\begin{matrix}a_1,\ldots,a_p\\ a_1-k,b_2,\ldots,b_q\end{matrix};z\right] = \sum_{l=0}^{k}\frac{(-1)^l(-k)_l(a_2)_l\ldots(a_p)_l z^l}{l!\,(a_1-k)_l(b_2)_l\ldots(b_q)_l}
$$
$$
\times{}_{p-1}F_{q-1}\left[\begin{matrix}a_2+l,\ldots,a_p+l\\ b_2+l,\ldots,b_q+l\end{matrix};z\right].
$$

`diff/hypergeom` implements the differentiation formula

$$
\frac{d}{dz}\,{}_pF_q\left[\begin{matrix}a_1,\ldots,a_p\\ b_1,\ldots,b_q\end{matrix};z\right] = \frac{a_1\ldots a_p}{b_1\ldots b_q}
$$
$$
\times{}_pF_q\left[\begin{matrix}a_1+1,\ldots,a_p+1\\ b_1+1,\ldots,b_q+1\end{matrix};z\right].
$$

`evalf/hypergeom` gives a numeric evaluation of a hypergeometric series with numeric argument and parameters by an algorithm which is not very sophisticated.

`series/hypergeom` gives the explicit series expansion, up to a certain term, of an hypergeometric expression.

5. SOME SUGGESTIONS FOR IMPROVEMENT

Here we have in mind a user of the second category (cf. section 3), who would like to use Maple as a special functions laboratory without the need of much further programming.

5.1. NONPOSITIVE PARAMETERS.

It should be possible to declare certain expressions to be nonnegative integers. As long as this is not possible within the Maple shell in general, this might be arranged by admitting an optional fourth argument $\{\mathtt{n1},\mathtt{n2},\ldots,\mathtt{ni}\}$ to `hypergeom` which should be the set of expressions (involving parameters of `hypergeom`) assumed to be in $\mathbf{Z}_+$. This would be helpful for looking up in a database explicitly summable series under some restriction (cf. Saalschütz's and Dougall's identity). It would also make possible the input into Maple of expressions

$$
\mathtt{hypergeom}([-\mathtt{n},\mathtt{a2},\ldots,\mathtt{ap}],[-\mathtt{N},\mathtt{b2},\ldots,\mathtt{bq}],\mathtt{z},\{\mathtt{n},\mathtt{N}-\mathtt{n}\})
$$

which should represent a hypergeometric series

$$
{}_pF_q\left[\begin{matrix}-n,a_2,\ldots,a_p\\ -N,b_2,\ldots,b_q\end{matrix};z\right] := \sum_{k=0}^{n}\frac{(-n)_k\,(a_2)_k\ldots(a_p)_k\,z^k}{(-N)_k\,(b_2)_k\ldots(b_q)_k\,k!},
$$
$$
0\le n\le N.
$$

This convention of allowing nonpositive lower parameters when they are majorized by nonpositive upper parameters is standard in the theory of hypergeometric functions, but not allowed in Maple. However, it is a very useful convention because many of the hypergeometric series met in nature have this form, in particular Krawtchouk polynomials and Hahn polynomials (essentially Clebsch-Gordan coefficients), which are orthogonal polynomials on a finite set.

In this way we might also introduce indefinite hypergeometric sums in hypergeometric notation:

$$\texttt{hypergeom}([-\texttt{n}, \texttt{a2}, \ldots, \texttt{ap}], [-\texttt{n}, \texttt{b2}, \ldots, \texttt{bq}], \texttt{z}, \{\texttt{n}\})$$

would mean the series

$${}_pF_q\left[\begin{matrix}-n, a_2, \ldots, a_p \\ -n, b_2, \ldots, b_q\end{matrix}; z\right] := \sum_{k=0}^{n} \frac{(a_2)_k \ldots (a_p)_k \, z^k}{(b_2)_k \ldots (b_q)_k \, k!} \, .$$

In the present situation Maple would let the upper and lower parameter $-n$ cancel each other, losing the upper summation boundary n.

Of course the procedure `simplify/gamma` should also be adapted: If `n` is nonnegative integer then `GAMMA(-n)` cannot be accepted in output.

One has to remain cautious with allowing lower parameters of hypergeometic functions to be nonpositive integers. For instance, ${}_2F_1(a, a+1/2; 2a; z)$ will not be continuous in a at the nonpositive integers and its explicit evaluation

$$2^{2a-1} \, (1-z)^{-1/2} \, (1+\sqrt{1-z})^{1-2a}$$

given by [6, 2.8(6)] will not be valid for $a \in -\mathbf{Z}_+$. Still, as was observed in a bug report at the Maple headquarters,

```
sum(binomial(2*n-k,k), k=0..n);
```

returns an answer based on conversion to ${}_2F_1(-n+1/2, -n; -2n; -4)$ and subsequent evaluation as if the expression were continuous in n.

5.2. TRANSFORMATION FORMULAS.

The present Maple routines are very much directed at simplification. The worker in special functions is often interested as well in transformation formulas. A few simple ones are:

$${}_2F_1(a, b; c; z) = (1-z)^{-a} \, {}_2F_1(a, c-b; c; z/(z-1))$$

(Pfaff-Kummer transformation formula),

$${}_2F_1(a, b; c; z) = (1-z)^{c-a-b} \, {}_2F_1(c-a, c-b; c; z)$$

(Euler's transformation formula).

The problem for computer algebra is that there is usually not an unambiguous choice of transformation formula. So Maple should give on request a list of possible transformations of a certain type of the input expression. Then the user can indicate with which of the alternatives he wants to proceed. It seems that such lists of transformation formulas will be much easier produced by the database approach than by algorithm.

5.3. SUBSTITUTIONS.

In computations with special functions one often wishes to substitute some transformation formula in some deeper level of a big expression. Therefore it is annoying that standard Maple routines only allow substitution at the first level of an expression. Better routines should be supplied and possibly also a user's interface where the user can indicate with the mouse to which part of an expression at the screen some substitution should apply.

5.4. MULTIPLE SUMS.

One of the techniques most frequently used in demonstrations of special functions identities is to pass by substitutions from single to double or multiple sums, fiddle around with the summation variables and simplify in the new summation order to a single sum. Although the ideas in such demonstrations are often conceptual, the technical details can be tedious and error prone. Computer algebra would be of much help here, but Maple does not have facilities for double summation.

5.5. ONLINE REFERENCES.

Maple should accompany the identities it produces, in particular the ones obtained from database, by a message stating the name of the identity, possibly also a reference, as soon as the user has set a certain `printlevel` or some other option made for this purpose.

6. DATABASE VERSUS ALGORITHM

In the examples of section 3 we have seen that Maple may use formula look-up from its own database or an algorithmic approach depending on the input. These two methods are dramatically different and, of course, the second method is much more glamorous for the computer algebrist. But for the user it is irrelevant which method has been used. He will be interested in (i) speed, (ii) correctness of the answer if an answer is produced, (iii) success in finding an answer if an answer is theoretically possible. Regarding these criteria the database approach may often be superior in speed, but a well-tested implementation of an algorithm will usually give more reliable answers than a database, for the same reason that formula books are usually not error-free. The algorithmic approach will usually also produce an answer in all cases where this is possible by theory, except that it may require a very long time.

In section 7 and 8 we will discuss the two successful algorithms by Gosper and Zeilberger, but here some further remarks about the balance between database and algorithm can already be made.

1. The algorithmic approach usually does not apply to evaluation of infinite sums. Still it is reasonable to conjecture that each explicitly summable infinite series is a limit case of an explicitly summable finite series (necessarily involving at least one more parameter). See the limit transitions mentioned after the examples of summation formulas in section 2.

2. Transformation formulas are harder to prove or to produce by algorithm than summation formulas. Zeilberger [14, §7.2] suggests an interesting method of demonstrating identities by showing that both sides satisfy the same recurrence relation and initial conditions. Still this method will require much interaction with the user and cannot be easily implemented in the black box manner as already has been done in Maple with Gosper's algorithm.

3. Even the database approach has some challenging aspects. Suppose that all known summation formulas from literature have been collected in a database then writing an adequate search algorithm will still be highly nontrivial.

4. The computer algebra program might become more intelligent in deciding in an early stage whether a given input expression is more suitable for algorithm or data base look-up. Maple's `sum` procedure is a little dumb in first always trying Gosper's algorithm until this has definitely failed.

Here we discuss Gosper's algorithm, of which the use in Maple was already shown by example in section 3. The fundamental reference is Gosper [9], see also Hayden & Lamagna [10].

Consider an indefinite summation

$$S(k_2) - S(k_1) := \sum_{k=k_1+1}^{k_2} a_k, \tag{7.1}$$

where the summand a_k is explicitly given and a_k/a_{k-1} is known to be rational in k. The indefinite sum $S(k)$ is determined up to a constant term and also solves the difference equation

$$S(k) - S(k-1) = a_k\,. \tag{7.2}$$

Gosper's algorithm decides whether $S(k)/S(k-1)$ is rational in k and finds $S(k)$ explicit if the quotient is rational.

First step. Write

$$\frac{a_k}{a_{k-1}} = \frac{p_k}{p_{k-1}}\,\frac{q_k}{r_k}, \tag{7.3}$$

where p_k, q_k, r_k are polynomials and $\gcd(q_k, r_{k+j}) = 1$ for $j \in \mathbf{Z}_+$. This is always possible.

Second step. Suppose $f(k)$ satisfies

$$p_k = q_{k+1}\,f(k) - r_k\,f(k-1). \tag{7.4}$$

Then

$$a_k = \frac{q_{k+1}}{p_k}\,f(k)\,a_k - \frac{q_k}{p_{k-1}}\,f(k-1)\,a_{k-1}\,, \tag{7.5}$$

so

$$S(k) := \frac{q_{k+1}}{p_k}\,f(k)\,a_k \tag{7.6}$$

satisfies (7.2) and hence (7.1).

Third step. Give an a priori bound $d = d(\{p_k, q_k, r_k\})$ for the degree of a solution $f(k)$ to (7.4) as follows.

a) $d := \deg(p_k) - \deg(q_{k+1} - r_k)$ if $\deg(q_{k+1} - r_k) \geq \deg(q_{k+1} + r_k)$.

b) if $\deg(q_{k+1} - r_k) \leq \deg(q_{k+1} + r_k) =: l$ then put $q_{k+1} - r_k = d_{l-1}\,k^{l-1} + \cdots$ and $q_{k+1} + r_k = e_l\,k^l + \cdots$.

   b1) $d := \deg(p_k) - l + 1$ if $-2d_{l-1}/e_l \notin \mathbf{Z}_+$.

   b2) $d := \max\{\deg(p_k) - l + 1, -2d_{l-1}/e_l\}$ if $-2d_{l-1}/e_l \in \mathbf{Z}_+$.

Fourth step. It can be proved that $S(k)/S(k-1)$ is rational if and only if $f(k)$ is a polynomial solution of (7.4) of degree $\leq d$.

Fifth step. Substitute $f(k) = \sum_{i=0}^{d} f_i k^i$ in (7.4) and solve for the f_i's. If a solution exists then the corresponding f will yield $S(k)$ by (7.6). Otherwise there is no rational $S(k)$.

Example. Let $\sum_{k=0}^{n} a_k$ be the indefinite case (2.7) of Saalschütz's summation. Then

$$a_k := \frac{(a)_k\,(b)_k}{(a+b+1)_k\,k!}\,.$$

This is extended to general complex k by substitution of gamma functions as in (2.2). We see that $a_{-1} = 0$ and

$$\frac{a_{k+1}}{a_k} = \frac{(a+k)\,(b+k)}{(a+b+k+1)\,(k+1)}\,,$$

so $q_k = (a+k)(b+k)$, $r_k = (a+b+k+1)(k+1)$, $p_k = 1$. (Note that we work here with rational functions not just over the field $\mathbf{C}$ but over the field of rational functions in a, b. The general considerations above remain valid.) We have to solve f from

$$1 = (a+k+1)\,(b+k+1)\,f(k) - (a+b+k+1)\,(k+1)\,f(k-1).$$

Then we are in case b2) of the third step above, with $q_{k+1}+r_k = 2\,k^2+\cdots$ and $q_{k+1}-r_k = 0\,k+\cdots$. So $d = 0$ and the corresponding solution is $f(k) = 1/(ab)$. Thus, by (7.6), $S(n)$ equals the right hand side of (2.7). Since $a_{-1} = 0$ we have $S(-1) = 0$, so $\sum_{k=0}^{n} a_k = S(n) - S(-1) = S(n)$ and the identity (2.7) is produced by Gosper's algorithm.

8. ZEILBERGER'S ALGORITHM

In this section we discuss Zeilberger's [14] algorithm which applies to the more common situation of summability of sums $\sum_{k=0}^{n} a_{n,k}$, where the summand depends on the upper summation boundary. We are dealing here with *definite* summability: for arbitrary m the sum $\sum_{k=0}^{m} a_{n,k}$ will not be summable. Zeilberger's clever trick reduces this situation to that of Gosper's algorithm.

Let us start with an indefinite summation (7.1) such that everything depends on an additional paramer n:

$$S(n,k_2) - S(n,k_1) = \sum_{k=k_1+1}^{k_2} a_{n,k}\,. \tag{8.1}$$

Suppose $a_{n+1,k}/a_{n,k}$ and $a_{n,k+1}/a_{n,k}$ are rational in n, k. In general, the corresponding quotients for $S(n,k)$ will not be rational in n, k and $S(n,k)$ will not have a nice explicit expression. However, $S(n,k)$ may satisfy some recurrence relation in n, in the simplest case this will take the form that $S(n+1,k) - s_0(n)\,S(n,k)$ has a nice explicit expression for some $s_0(n)$ which is rational in n.

As an Ansatz, conclude from (8.1) that

$$\begin{aligned}\big(S(n+1,k_2) - s_0(n)\,S(n,k_2)\big) - \big(S(n+1,k_1) - s_0(n)\,S(n,k_1)\big)& \\ = \sum_{k=k_1+1}^{k_2} \left(a_{n+1,k} - s_0(n)\,a_{n,k}\right),&\end{aligned}$$

where $s_0(n)$ is arbitrarily rational. Thus we have (7.1) with

$$a_k := a_{n+1,k} - s_0(n)\,a_{n,k} \tag{8.2}$$

and

$$S(k) := S(n+1,k) - s_0(n)\,S(n,k) \tag{8.3}$$

and a_k/a_{k-1} is rational in k over the field of rational functions in n and s_0. Now go through the various steps of the previous section, keeping $s_0(n)$ as an unknown.

First we can write (7.3) as

$$\frac{a_k}{a_{k-1}} = \frac{(p_{n,k} - \tilde{p}_{n,k}\, s_0(n))\, q(n,k)}{(p_{n,k-1} - \tilde{p}_{n,k-1}\, s_0(n))\, r(n,k)}\,, \tag{8.4}$$

where $p_{n,k}$, $\tilde{p}_{n,k}$, $q(n,k)$, $r(n,k)$ are polynomials in k with coefficients rational in n and $\gcd(q(n,k), r(n,k+j)) = 1$ for $j \in \mathbf{Z}_+$. (It is crucial that $s_0(n)$ happens to occur in (8.4) only in this particular way.)

Next we try to solve (7.4), which takes here the form

$$p_{n,k} - \tilde{p}_{n,k}\, s_0(n) = q(n,k+1)\, f(n,k) - r(n,k)\, f(n,k-1). \tag{8.5}$$

We use the a priori bound d of the third step of section 7 for the degree of $f(n,k)$ as a polynomial in k. The coefficients of $f(n,k)$, up to degree d, will be unknown functions of n and these, together with $s_0(n)$, will appear as unknowns in a linear system of equations (obtained from (8.5)) over the field of rational numbers. If the system can be solved then (7.6) takes the form of the desired two-terms recurrence relation

$$S(n+1,k) - s_0(n)\, S(n,k) = \frac{q(n,k+1)}{p_{n,k} - \tilde{p}_{n,k}\, s_0(n)}\, f(n,k)\, a_k\,.$$

In order to solve this recurrence relation, we need initial conditions. These are available, for instance, if it is given that $a_{n,k} = 0$ whenever $n \in \mathbf{Z}_+$ and $k = -1, -2, \dots$ or $n+1, n+2, \dots$. Then it will follow that

$$\sum_{k=0}^{n} a_{n,k} = s_0(0)\, s_0(1) \dots s_0(n-1)\, a_{0,0}\,. \tag{8.6}$$

Example. Let $\sum_{k=0}^{n} a_k$ be the definite case (2.6) of Saalschütz's summation. Then

$$a_{n,k} := \frac{(a)_k\, (b)_k\, (-n)_k}{(c)_k\, (1+a+b-c-n)_k\, k!}\,,$$

which is extended to general complex k by expressing everything in terms of gamma functions. We see that $a_{n+1,k}/a_{n,k}$ and $a_{n,k+1}/a_{n,k}$ are rational in n, k and that $a_{n,k} = 0$ if $n \in \mathbf{Z}_+$ and $k = -1, -2, \dots$ or $n+1, n+2, \dots$. We obtain (8.4) with

$$\begin{aligned}
p_{n,k} &:= (-n-1)\, (a+b-c-n+k),\\
\tilde{p}_{n,k} &:= (a+b-c-n)\, (-n+k-1),\\
q_{n,k} &:= (a+k-1)\, (b+k-1)\, (-n+k-2),\\
r_{n,k} &:= (c+k-1)\, (a+b-c-n+k)\, k.
\end{aligned}$$

Thus we have to solve

$$\begin{aligned}
&(-n-1)\, (a+b-c-n+k)\\
&\quad - (-n+k-1)\, (a+b-c-n)\, s_0(n)\\
&= (a+k)\, (b+k)\, (-n+k-1)\, f(n,k)\\
&\quad - (a+b-c-n+k)\, (c+k-1)\, k\, f(n,k-1).
\end{aligned} \tag{8.7}$$

Again we are in the case b2) of the third step of section 7 with $q(n,k+1)+r(n,k) = 2\,k^3+\cdots$ and $q(n,k+1)-r(n,k) = 0\,k^2+\cdots$. So $d=0$ and we have to look for a solution $f(n)=f(n,k)$ of (8.7) not depending on k. Collect terms in (8.7) of first respectively zero degree in k. We obtain the system of two equations

$$\begin{cases} (-n-1)\,(a+b-c-n)\,(1-s_0(n)) = (-n-1)\,ab\,f(n), \\ \quad -n-1-(a+b-c-n)\,s_0(n) \\ = \big(ab+(a+b)(-n-1)-(a+b-c-n)\,(c-1)\big)\,f(n) \end{cases}$$

in the unknowns $f(n)$ and $s_0(n)$. It can be solved with solution

$$f(n) = \frac{1}{c+n}\,, \quad s_0(n) = \frac{(c+n-a)\,(c+n-b)}{(c+n)\,(c+n-a-b)}\,.$$

Now we obtain (2.6) from (8.6).

Zeilberger [14] presents a similar method for obtaining higher order recurrences. Just replace (8.3) by

$$S(k) := S(n+1,k) - \sum_{j=0}^{l} s_j(n)\,S(n-j,k),$$

where the $s_j(n)$ are unknowns. By the theory of holonomic systems, Zeilberger [13] shows that in the generic case there is always a value of l for which the method will succeed. For explicitly given orthogonal polynomials and $l=1$ this is a powerful method to get the three-term recurrence relation for the orthogonal polynomials by computer algebra.

There is one caveat. Even if the left hand side of (8.6) is explicitly summable, Zeilberger's algorithm with $l=0$ may fail, but will for instance succeed with $l=1$. According to Zeilberger [14, §7.1] this will happen in only very rare cases and I do not yet know an example of this phenomenon.

Zeilberger has implemented his algorithm in a Maple procedure in a funny way, such that the output is a ready-made paper (cf. for instance Ekhad & Zeilberger [5]) proving some explicit summation or recurrence formula, together with an elementary proof ((7.5) and (8.2)).

9. FURTHER PERSPECTIVES

There are some aspects of special functions not covered in this paper, but still suitable for computer algebra. First there is the generalization of hypergeometric series to *q-hypergeometric series*: a series $\sum_{k=0}^{\infty} c_k$ with c_{k+1}/c_k rational in q^k rather than k. There is a rich and still fast developing theory of q-special functions, including many summation theorems, suitable for both the database and the algorithmic approach. See Gasper & Rahman [8] for the basic theory and Gasper [7] for some computer algebra experiments in Mathematica.

As a second item there are the generalizations to several variables of hypergeometric and q-hypergeometric series. There is not yet a unification of the several generalizations, which range from very classical ones like the Appell and Lauricella hypergeometric functions to very recent ones associated to arbitrary root systems (Heckman, Opdam, Macdonald), but availability of computer algebra packages supporting these generalizations would be welcome.

A third item is the Askey tableau of hypergeometric orthogonal polynomials and its q-analogues, the Askey-Wilson polynomials, cf. Askey & Wilson [3]. The big complex of formulas associated with these polynomials should become available in the context of some computer algebra package.

Some aspects not discussed in this paper concern the numerical computation and the graphics of special functions. Neither have I made any comparisons with other programs like Reduce, Macsyma, Mathematica and Scratchpad. Hopefully this can be picked up (by someone else?) in some next issue.

Finally I would suggest that a better documentation becomes available about existing computer algebra routines for special functions which are not a standard part of the big programs as Maple etc. Maybe this can be a task for the recently founded special interest group on Orthogonal Polynomials and Special Functions within SIAM (Society for Industrial and Applied Mathematics).

REFERENCES

[1] M. Abramowitz and I. Stegun, *Handbook of mathematical functions*, Dover, 1965.

[2] R. Askey, *How can mathematicians and mathematical historians help each other?*, in *History and Philosophy of modern mathematics* (W. Aspray and Ph. Kitcher, eds.), University of Minnesota Press, 1988, pp. 201–217.

[3] R. Askey and J. Wilson, *Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials*, Mem. Amer. Math. Soc. **54** (1985) no. 319.

[4] W. N. Bailey, *Generalized hypergeometric series*, Cambridge University Press, 1935.

[5] S. B. Ekhad and D. Zeilberger, *A 21^{st} century proof of Dougall's hypergeometric sum identity*, J. Math. Anal. Appl. 147 (1990), 610–611.

[6] A. Erdélyi e.a., *Higher Transcendental Functions, Vol. I*, McGraw-Hill, 1953.

[7] G. Gasper, *Using symbolic computer algebraic systems to derive formulas involving orthogonal polynomials and other special functions*, in *Orthogonal polynomials: theory and practice* (P. Nevai, ed.), NATO ASI Series C, Vol. 294, Kluwer, 1990, pp. 163–179.

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

[9] R. W. Gosper, Jr., *Decision procedure for indefinite hypergeometric summation*, Proc. Nat. Acad. Sci. USA **75** (1978), 40–42.

[10] M. B. Hayden and E. A. Lamagna, *Summation of binomial coefficients using hypergeometric functions*, in *Proceedings of the 1986 Symposium on Symbolic and Algebraic Computation, Symsac '86* (B. W. Char, ed.), ACM, 1986.

[11] J. Riordan, *Combinatorial identities*, Wiley, 1968.

[12] L. J. Slater, *Generalized hypergeometric functions*, Cambridge University Press, 1966.

[13] D. Zeilberger, *A holonomic systems approach to special functions identities*, J. Comput. Appl. Math. 32 (1990), 321-368.

[14] D. Zeilberger, *The method of creative telescoping*, J. Symb. Comput. 11 (1991), 195–204.