Summation in Maple

Wolfram Koepf*

Introduction

Maple's `sum` command is a general purpose procedure to calculate different types of sums. Algorithms for the computation of both indefinite and definite sums are available.

Roberto Pirastu [1] gave an excellent overview about rational summation, and we take his article as a starting point for ours, and thus will not focus on the rational case.

We will instead discuss three important methods for more general summation,

1. Gosper's algorithm [2] (and an extension of the author [3]) for indefinite summation,
2. Zeilberger's algorithm [4]–[6] (and an extension of the author [3]) for definite summation,
3. and the representation of sums as hypergeometric functions.

We further deal with some particularly important simplification issues, and we present examples of new implementations in Maple that are available with Release V.4.

The `sum` command of the new Maple release contains considerable improvements to the rational function case (done by Jacques Carette), so that for example

```
> sum((k^2+3)/(k^2*(k+2)^3*(k+3)^2*
>      (k^2+2)*(k^2+2*k+3)^2),k);
```

(see [1]) can be computed now. To free the `sum` command from doing number crunching, the new release contains the commands `add` and `mul` (done by Michael Monagan), that should be used for the calculation of finite sums and products.

Finally, the `sum` command includes my improvements of the Gosper algorithm, and a version of Zeilberger's algorithm. These algorithms are also directly available through the package `sumtools`:

```
> with(sumtools);
```

$$[\mathit{Hypersum}, \mathit{Sumtohyper}, \mathit{extended\_gosper}, \mathit{gosper}, \mathit{hyperrecursion}, \mathit{hypersum}, \mathit{hyperterm}, \mathit{simpcomb}, \mathit{sumrecursion}, \mathit{sumtohyper}]$$

*Konrad-Zuse-Zentrum für Informationstechnik Berlin, Heilbronner Str. 10, D-10711 Berlin, koepf@zib-berlin.de

Gosper's algorithm

Gosper's algorithm [2] deals with the problem of *indefinite summation*: Given a_k, one wants to find s_k such that

$$a_k = s_{k+1} - s_k \ ,$$

i.e., an *antidifference* of a_k. In the affirmative case, any sum with summand a_k can be calculated by an evaluation of s_k at the boundary points

$$\sum_{k=m}^{n} a_k = s_{n+1} - s_m$$

if a_k does not depend on m and n, similarly to the integration case when an antiderivative is known.

Gosper's algorithm is a decision procedure which finds an antidifference s_k if it is a *hypergeometric term*, i.e., if

$$\frac{s_{k+1}}{s_k}$$

is a rational function in k. In this case, the input function a_k must itelf be a hypergeometric term.

Gosper's algorithm either returns a hypergeometric term antidifference s_k of a_k or the statement `'no hypergeometric term antidifference exists'`.

Gosper's algorithm is accessible via the two argument version of Maple's `sum` command some examples of which are

```
> sum(k*k!,k);
```

$$k!$$

```
> sum(binomial(k,n),k);
```

$$\frac{(k-n)\,\mathrm{binomial}(k,n)}{n+1}$$

```
> sum((-1)^(k+1)*(4*k+1)*(2*k)!/
>      (k!*4^k*(2*k-1)*(k+1)!),k);
```

$$-2\,\frac{(k+1)\,(-1)^{(k+1)}\,(2\,k)!}{k!\,4^k\,(2\,k-1)\,(k+1)!}$$

which leads to

```
> sum((-1)^(k+1)*(4*k+1)*(2*k)!/
>      (k!*4^k*(2*k-1)*(k+1)!),k=1..infinity);
```

$$1$$

Therefore the latter infinite sum—which solves a question raised in SIAM Review [7]—was computed by an application of Gosper's algorithm, and an appropriate limit computation.

Which algorithms are used by `sum`, can be checked via `infolevel`, (because of `sum`'s remember table, you must use a new Maple session):

```
>  infolevel[sum]:=3:
>  sum((-1)^(k+1)*(4*k+1)*(2*k)!/
>     (k!*4^k*(2*k-1)*(k+1)!),k=1..infinity);

sum/infinite:   infinite summation
sum/indefnew:   indefinite summation
sum/extgosper:    applying Gosper algorithm to
                a(k):=(-1)^(k+1)*(4*k+1)*(2*k)!/
                k!/(4^k)/(2*k-1)/(k+1)!
sum/gospernew:    a(   k   )/a(   k   -1):=
                -1/2*(4*k+1)/(4*k-3)/(k+1)*(2*k-3)
sum/gospernew:    Gosper's algorithm applicable
sum/gospernew:    p:=   4*k+1
sum/gospernew:    q:=   -2*k+3
sum/gospernew:    r:=   2*k+2
sum/gospernew:    degreebound:=   0
sum/gospernew:    solving equations to find f
sum/gospernew:    Gosper's algorithm successful
sum/gospernew:    f:=   -1
sum/indefnew:   indefinite summation finished
```

$$1$$

In the case of rational function input, Maple does not use Gosper's algorithm, but utilizes different methods, as discussed in [1], although Gosper's algorithm is always successful if a rational antidifference exists.

We mention that Gosper's algorithm needs a *dispersion* computation (see [1]). The *dispersion set* of the two polynomials q_k, and r_k is given by

$$J := \{j \in \mathbb{N}_0 \mid \gcd(q_k, r_{k+j}) \not\equiv 1\} \ .$$

The Maple V.3 implementation uses resultants to compute the dispersion set, an approach which is inherently very time consuming:

```
>  term:=3^k*(3*k^2+2*a*k-4*k-2-a)/
>  ((2*k+2+a)*(2*k+a)*(k+1)*k)*binomial(n,k):
>  term:=normal(subs(k=k+3,term)-term):
>  TIME:=time(): sum(term,k): time()-TIME;
```

$$926.700$$

In the new implementation, a faster approach using rational factorization is used ([8], [9]):

```
>  term:=3^k*(3*k^2+2*a*k-4*k-2-a)/
>  ((2*k+2+a)*(2*k+a)*(k+1)*k)*binomial(n,k):
>  term:=normal(subs(k=k+3,term)-term):
>  TIME:=time(): sum(term,k): time()-TIME;
```

$$21.520$$

We give a brief review of Gosper's algorithm:

- **Input a_k:** Assume
$$\frac{a_{k+1}}{a_k} = \frac{b_k}{c_k} \ , \qquad b_k, c_k \text{ polynomials in } k \ .$$
If the summand a_k is given, one needs to determine the polynomials b_k and c_k.

- **Polynomial part:** Find a representation
$$\frac{b_k}{c_k} = \frac{p_{k+1}}{p_k}\frac{q_{k+1}}{r_{k+1}}$$
where p_k, q_k, r_k are polynomials in k, for which
$$\gcd(q_k, r_{k+j}) = 1 \qquad \text{for all } j \in \mathbb{N}_0 \ .$$
The polynomial p_k corresponds to the *polynomial part*, and (q_k, r_k) to the *factorial part* of a_k. This step needs a dispersion computation.

- **Crucial observation**: f_k, defined by
$$f_k := \frac{s_{k+1}}{a_{k+1}}\frac{p_{k+1}}{r_{k+1}}$$
(s_k denoting the unknown antidifference) is rational, but because of the above gcd-condition, f_k is in fact a polynomial in k which satisfies the inhomogeneous linear recurrence equation
$$p_k = q_{k+1}\, f_k - r_k\, f_{k-1} \ . \tag{1}$$

- **Degree bound**: An upper bound for the degree of f_k is determined by the following algorithm.
  1. Let $n = \deg(q_{k+1} + r_k)$
  2. If $n \leq \deg(q_{k+1} - r_k)$, then $\deg f_k = \deg p_k - \deg(q_{k+1} - r_k)$.
  3. Otherwise let a be the coefficient of k^n in the polynomial $q_{k+1} + r_k$, and b be the coefficient of k^{n-1} in $q_{k+1} - r_k$.
     If $-2b/a \notin \mathbb{N}_0$ then
     $$\deg f_k = \deg p_k - n + 1 \ .$$
     Otherwise
     $$\deg f_k \leq \max\{-2b/a, \deg p_k - n + 1\}.$$

  If $\deg f_k < 0$ then return `'no hypergeometric term antidifference exists'`.

- **Calculate f_k**: Given the maximal degree of f_k, by equating coefficients in Equation (1), this is pure linear algebra. If no such f_k exists, return `'no hypergeometric term antidifference exists'`.

- **Output:** $s_k = \frac{r_k}{p_k}\, f_{k-1}\, a_k$.

Note that the degree bound is found by an inspection of Equation (1).

We give an example of Gosper's algorithm. Assume $a_k = \binom{n}{k}$, i.e., we want to know whether or not the sum

$$\sum_{k=0}^{m} \binom{n}{k}$$

has a hypergeometric term representation for arbitrary m, therefore extending the binomial theorem.

In the given case, since

$$\frac{a_{k+1}}{a_k} = \frac{n-k}{k+1}\,,$$

we get $p_k = 1$, $q_k = n-k+1$, and $r_k = k$. Gosper's degree bound for f_k leads to the value -1, therefore *proving* that the antidifference of a_k is *not* expressible as hypergeometric term.

We change the example a little bit, and consider now $a_k = (-1)^k \binom{n}{k}$, the alternating binomial coefficients. In this case, we get $p_k = 1$, $q_k = k-n-1$, and $r_k = k$, and the degree bound for f_k gives 0. So there is a chance for a hypergeometric term antidifference! We take the generic polynomial $f_k = c$ for some constant c, and substitute this into Equation (1). Hence we get the identity

$$1 = (k-n)\,c - k\,c = -nc$$

with the obvious solution $c = -1/n$, so that $f_k = -1/n$, and finally

$$s_k = \frac{r_k}{p_k}\, f_{k-1}\, a_k = -\frac{k}{n}\, a_k = -\frac{k}{n}\,(-1)^k \binom{n}{k}\,.$$

Therefore, Maple's `sum` command yields

```
> sum((-1)^k*binomial(n,k),k);
```

$$-\frac{k\,(\,-1\,)^k\,\mathrm{binomial}(\,n,k\,)}{n}$$

The `sum` command (and its Gosper implementation) in Maple V.3 had, however, several severe problems. We give some examples.

The first example has a rational antidifference:

```
> sum(1/(k+1)-1/k,k);
```

$$\psi(\,k+1\,) - \psi(\,k\,)$$

This fact is not realized since the `sum` command uses linearity, and treats each summand separately; that is the default if the first argument of `sum` is a finite sum. This situation can be resolved by the input

```
> sum(normal(1/(k+1)-1/k),k);
```

$$\frac{1}{k}$$

which is not a satisfactory solution since not every user might be aware of the problem. Algorithms for indefinite summation like Gosper's are highly non-linear, so that linearity should be avoided. The new version of the `sum` command takes care of this situation.

For more difficult examples, `normal` may not help, like in the case

```
> sum(normal(
> binomial(n+1,k)/2^(n+1)-binomial(n,k)/2^n),
> k);
```

$$\sum_{k} (\mathrm{binomial}(\,n+1,k\,)\,2^n - \mathrm{binomial}(\,n,k\,)\,2^{(\,n+1\,)})\,/(\,2^{(\,n+1\,)}\,2^n)$$

Here linearity is not the issue since the normalized input is *not* a sum. Instead, *simplification* of a_{k+1}/a_k fails to decide that this term ratio is rational. Therefore, the fact that Gosper's algorithm is a decision procedure, is completely lost, and the `infolevel`-message `Gosper's algorithm fails` is erroneous!

In the new release, one has

```
> sum(
> binomial(n+1,k)/2^(n+1)-binomial(n,k)/2^n,
> k);
```

$$-\frac{k}{2\,k-1-n}\left(\frac{\mathrm{binomial}(\,n+1,k\,)}{2^{(\,n+1\,)}} - \frac{\mathrm{binomial}(\,n,k\,)}{2^n}\right)$$

To give you an idea what the problem is, we consider the following simplification issues that are connected with the above failure: With Maple V.3, the simplification of factorial and Gamma function as well as of power terms did not work adequately. For example

```
> a:=simplify(factorial(k)+factorial(k+1));
```

$$a := \Gamma(\,k+1\,) + \Gamma(\,k+2\,)$$

```
> ratio:=simplify(subs(k=k+1,a)/a);
```

$$\mathit{ratio} := \frac{\Gamma(\,k+2\,) + \Gamma(\,k+3\,)}{\Gamma(\,k+1\,) + \Gamma(\,k+2\,)}$$

The fact that the first `simplify` command keeps two different `GAMMA` terms that are rational multiples of each other, makes it impossible in the second step to decide that a_{k+1}/a_k is rational. The new release will give instead

```
> a:=simplify(factorial(k)+factorial(k+1));
```

$$a := (k+2)\,\Gamma(\,k+1\,)$$

```
> ratio:=simplify(subs(k=k+1,a)/a);
```

$$\mathit{ratio} := \frac{(\,k+1\,)\,(\,k+3\,)}{k+2}$$

This new simplification code is independent of the new summation features, and was done by Mike Monagan.

Similarly, for powers, Maple gives

```
> a:=simplify(m^k+m^(k+1));
```

$$a := m^k + m^{(k+1)}$$

```
> ratio:=simplify(subs(k=k+1,a)/a);
```

$$ratio := \frac{m^{(k+1)} + m^{(k+2)}}{m^k + m^{(k+1)}}$$

In [3], we presented an extension of Gosper's algorithm for the case that a_{k+l}/a_k is rational for some $l \in \mathbb{N}, l > 1$. This extension is also covered by Maple's new `sum` command, e.g.

```
> sum(k*(k/3)!,k);
```

$$3\left(\frac{1}{3}k\right)! + 3\left(\frac{1}{3}k + \frac{1}{3}\right)! + 3\left(\frac{1}{3}k + \frac{2}{3}\right)!$$

```
> sum(binomial(k/2,n),k);
```

$$\frac{\left(\frac{1}{2}k - n\right)\,\mathrm{binomial}\left(\frac{1}{2}k, n\right)}{n+1} + \left(\frac{1}{2}k + \frac{1}{2} - n\right)\,\mathrm{binomial}\left(\frac{1}{2}k + \frac{1}{2}, n\right) / (n+1)$$

The new Gosper implementation is directly available loading the package `sumtools`:

```
> gosper(1/(1-k^2),k);
```

$$-\frac{1}{2}\,\frac{(k+1)(2k-1)}{k(1-k^2)}$$

```
> extended_gosper(k*(k/2)!,k);
```

$$2\left(\frac{1}{2}k\right)! + 2\left(\frac{1}{2}k + \frac{1}{2}\right)!$$

The function `simpcomb`, loaded with `sumtools`, simplifies any factorial-Γ-binomial input by conversion in Γ-notation, at the same time deciding whether or not it is rational:

```
> simpcomb(
> (binomial(n+1,k+1)-binomial(n,k+1))/
> (binomial(n+1,k)-binomial(n,k)));
```

$$-\frac{k-1-n}{k}$$

For the previous power expression, we get

```
> a:=simplify(m^k+m^(k+1));
```

$$a := m^k + m^{(k+1)}$$

```
> simpcomb(subs(k=k+l,a)/a);
```

$$m^l$$

Another reason why Maple's `simplify` command does not simplify some expressions that `simpcomb` does is because some assumptions are needed to make those simplifications and `simpcomb` makes those implicitly.

Many more examples of non-trivial applications of Gosper's algorithm (in particular examples for the Wilf-Zeilberger method [10]) are given in [3].

Zeilberger's algorithm

Zeilberger's algorithm [4]–[6] deals with *definite sums*. Here we mean sums of the form

$$\Sigma(n) = \sum_{k\in\mathbb{Z}} F(n,k) \;, \tag{2}$$

the sum to be taken with respect to all $k \in \mathbb{Z}$. In particular, this covers sums of the type

$$\sum_{k=k_1}^{k_2} F(n,k)$$

if $F(n,k) = 0$ for $k < k_1$ and $k > k_2$, e.g.

$$\Sigma(n) = \sum_{k\in\mathbb{Z}} \binom{n}{k} = \sum_{k=0}^{n} \binom{n}{k} \;.$$

Zeilberger's algorithm applies if $F(n,k)$ is a hypergeometric term with respect to both n and k. It generates a *holonomic recurrence equation*, i.e. a homogeneous linear recurrence equation with polynomial coefficients, for $\Sigma(n)$, given by Equation (2). If the recurrence equation is first order, then—if n is assumed to be an integer—$\Sigma(n)$ is easily converted to a hypergeometric term.

Here is a brief description of Zeilberger's algorithm:

- **Iteration:** Iterate on J: Set

  $$a_k := F(n,k) + \sum_{j=1}^{J} \sigma_j(n)\, F(n+j,k)$$

  with as yet undetermined variables σ_j depending on n, but not depending on k.

- **Gosper algorithm**: Apply a simple adaption of Gosper's algorithm to a_k; in the last step, solve a linear system for the coefficients of f_k, and at the same time for the unknowns σ_j $(j = 1, \ldots, J)$. In the affirmative case, Gosper's algorithm finds $G(n,k)$ with

  $$G(n,k+1) - G(n,k) = a_k \;.$$

- **Output**: By summation:

$$\Sigma(n) + \sum_{j=1}^{J} \sigma_j(n)\, \Sigma(n+j) = 0$$

for $\Sigma(n)$, given by Equation (2).

Note that there is an upper bound for the order J of the resulting recurrence equation for input of special type (see [11], § 5.8), so that the algorithm terminates.

As an example, we consider the sum

$$\Sigma(n) := \sum_{k \in \mathbb{Z}} \binom{n}{k}$$

with $F(n,k) = \binom{n}{k}$. If we start with $J = 0$, then the whole procedure is Gosper's, and yields no antidifference as we saw earlier. Therefore, we try $J = 1$. Then we have

$$a_k = \binom{n}{k} + \sigma_1 \binom{n+1}{k} ,$$

hence

$$\frac{a_{k+1}}{a_k} = \frac{(n+1-k)\,(n-k+\sigma_1 n+\sigma_1)}{(n+1-k+\sigma_1 n+\sigma_1)\,(k+1)} . \qquad (3)$$

The dispersion calculation shows that we have $p_k = n+1-k+\sigma_1 n+\sigma_1$, $q_k = n+2-k$, and $r_k = k$ (notice the shift of the factors in numerator and denominator of (3)). The degree bound for f_k turns out to be 0. Substituting the generic polynomial $f_k = c$ in Equation (1) yields the identity

$$n+1-k+\sigma_1 n+\sigma_1 = (n+1-k\,)c - k\,c$$

and equating coefficients of like powers of k gives the linear equations

$$\begin{aligned} -1+2\,c &= 0 \\ n+1+\sigma_1 n+\sigma_1-(n+1)\,c &= 0 \end{aligned}$$

that we solve with respect to the unknowns $\{c, \sigma_1\}$, with the solution

$$\{c = 1/2, \sigma_1 = -1/2\} \ .$$

Therefore $f_k = 1/2$, but the more important information is $\sigma_1 = -1/2$ which leads us to the discovery of the recurrence equation

$$\Sigma(n) - \frac{1}{2}\Sigma(n+1) = 0 \ .$$

By $\Sigma(0) = 1$, we have $\Sigma(n) = 2^n$.

Zeilberger's algorithm was not implemented in Maple V.3. There is a package by Zeilberger [6] supporting only input of special form, and one by Koornwinder [8]. The latter was designed for hypergeometric input only, but it introduced the resultant-free dispersion computation, and served as the starting point of our implementation [3].

With Maple V.3, instead, in several instances definite sums could be explicitly calculated using a conversion to hypergeometric form

```
> sum(binomial(n,k),k=0..n);
```

$$2^n$$

```
> sum(binomial(n,k)^2,k=0..n);
```

$$\frac{\Gamma(\,1+2\,n\,)}{\Gamma(\,n+1\,)^2}$$

were computed by this method, whereas the corresponding infinite sum

```
> sum(binomial(n,k),k=-infinity..infinity);
```

$$\sum_{k=-\infty}^{\infty} \mathrm{binomial}(\,n,k\,)$$

could not be solved.

The new Maple release includes an implementation of Zeilberger's algorithm. To use this implementation, the user must make an assumption of the form

```
> assume(n,integer);
```

to declare an integer variable, n, say. For infinite sums this is essential information to be able to calculate initial values which, together with the holonomic recurrence equation, may yield a hypergeometric term result.

Having done this, we get e.g.[1]

```
> sum(binomial(n,k)^2,k=-infinity..infinity);
```

$$\frac{\Gamma(2\,\mathrm{n}+1)}{(\Gamma(\,\mathrm{n}+1))^2}$$

```
with assumptions on n
```

and also

```
> sum((-1)^k*binomial(n,k)^2,
> k=-infinity..infinity);
```

$$\begin{cases} 4\,\dfrac{\Gamma(\,n\,)\,\sqrt{(\,-1\,)^n}}{n\,\Gamma\,(n/2)^2} & \mathrm{irem}(n,2)=0 \\ 0 & \mathrm{irem}(n,2)=1 \end{cases} \qquad (4)$$

```
with assumptions on n
```

[1] with `interface(showassumed=2);`

Here, the resulting term has different representations for even and odd n, and is given by Maple's `piecewise` function.

With the `sumtools` package, one can use the algorithm directly: The statement

```
> sumrecursion(
> (-1)^k*binomial(n,k)^2,k,s(n));
```

calculates the recurrence equation

$$4\,(n-1)\,s_{n-2} + n\,s_n = 0$$

satisfied by

$$s_n = \sum_{k=-\infty}^{\infty} (-1)^k \binom{n}{k}^2 = \sum_{k=0}^{n} (-1)^k \binom{n}{k}^2 ,$$

which can be solved using the initial values $s_0 = 1$ and $s_1 = 0$. This is, how (4) is obtained.

The result

```
> sum(binomial(n,k)^2*binomial(n+k,k)^2,
> k=-infinity..infinity);
```

$$\begin{aligned}&\mathrm{RESol}\Big(\Big\{(\,n-1\,)^3\,\_\mathrm{F}(\,n-2\,)-\\ &\quad(\,2\,n-1\,)\,(\,17\,n^2-17\,n+5\,)\\ &\quad\_\mathrm{F}(\,n-1\,)+\_\mathrm{F}(\,n\,)\,n^3=0,\\ &\quad\_\mathrm{F}(\,0\,)=1,\_\mathrm{F}(\,1\,)=5\Big\}\,,\_\mathrm{F}(\,n\,)\Big)\end{aligned}$$

with assumptions on n

on the other hand, shows that no hypergeometric type solution was found, and therefore the recurrence equation and initial values of

$$\_\mathrm{F}\,(n) = \sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 ,$$

the so-called *Apéry numbers*, are returned (see [12]). This recurrence equation played an important role in Apéry's proof of the irrationality of

$$\zeta(3) = \sum_{k=1}^{\infty} \frac{1}{k^3} .$$

To present another example, we use this new feature of the `sum` command to generate three-term recurrence equations satisfied by families of orthogonal polynomials: The discrete Charlier polynomials, e.g., have a hypergeometric representation ([13], 2.7.3.1)

$$\begin{aligned} c_n^{(\mu)}(x) &= {}_2F_0\big(\,-n,\;-x\;\big|-1/\mu\big)\\ &= \sum_{k\in\mathbb{Z}} \binom{n}{k}\binom{x}{k}\,k!\left(-\frac{1}{\mu}\right)^k\end{aligned}$$

and therefore

```
> sum(
> binomial(n,k)*binomial(x,k)*k!*(-1/mu)^k,
> k=-infinity..infinity);
```

$$\begin{aligned}&\mathrm{RESol}\Big(\Big\{\_\mathrm{F}(\,1\,)=1-\frac{x}{\mu},\\ &\quad(\,n-1\,)\,\_\mathrm{F}(\,n-2\,)\\ &\quad+(\,x-n+1-\mu\,)\,\_\mathrm{F}(\,n-1\,)\\ &\quad+\_\mathrm{F}(\,n\,)\,\mu=0,\_\mathrm{F}(\,0\,)=1\Big\}\,,\\ &\quad\_\mathrm{F}(\,n\,)\Big)\end{aligned}$$

with assumptions on n

gives the recurrence equation with respect to n. If we wish to calculate the recurrence equation with respect to x, we enter

```
> n:='n': assume(x,integer):
> sum(
> binomial(n,k)*binomial(x,k)*k!*(-1/mu)^k,
> k=-infinity..infinity);
```

$$\begin{aligned}&\mathrm{RESol}\Big(\Big\{\_\mathrm{F}(\,0\,)=1,(\,x-1\,)\,\_\mathrm{F}(\,x-2\,)\\ &\quad-(\,x-n-1+\mu\,)\,\_\mathrm{F}(\,x-1\,)\\ &\quad+\_\mathrm{F}(\,x\,)\,\mu=0,\_\mathrm{F}(\,1\,)=1-\frac{n}{\mu}\Big\}\,,\\ &\quad\_\mathrm{F}(\,x\,)\Big)\end{aligned}$$

with assumptions on n

Hypergeometric Notation

If all else fails and no closed form solution is found, it may be helpful to give a hypergeometric representation of sums. This is done by the `convert/hypergeom` procedure, for example for the Apéry numbers

```
> sum(binomial(n,k)^2*binomial(n+k,k)^2,
> k=0..n);
```

$$\mathrm{hypergeom}(\,[\,-n,-n,n+1,n+1\,],[\,1,1,1\,],-1\,)$$

The `sumtools` package contains a procedure `sumtohyper` which does the conversion of an infinite sum into hypergeometric notation. It uses the `simpcomb` procedure for simplifications, and is mightier than the Maple V.3 `convert/hypergeom` procedure. However, the `convert/hypergeom` in Maple V.4 gives similar results.

As an example, we consider the Legendre polynomials $P_n(x)$, given by

$$P_n(x) := \sum_{k=0}^{n} \binom{n}{k} \binom{-n-1}{k} \left(\frac{1-x}{2}\right)^k . \qquad (5)$$

We get

```
> legendreterm:=binomial(n,k)*
> binomial(-n-1,k)*((1-x)/2)^k:
> sumtohyper(legendreterm,k);
```

$$\mathrm{hypergeom}\left([-n, n+1], [1], \frac{1}{2} - \frac{1}{2}x\right)$$

which is the hypergeometric equivalent of Equation (5). Note that we can easily derive further interesting hypergeometric representations for m-fold differences of successive Legendre polynomials:

```
> sumtohyper(subs(n=n+1,legendreterm)-
> legendreterm,k);
```

$$(-n + n\,x - 1 + x)\,\mathrm{hypergeom}\left([n+2, -n],\right.$$
$$\left.[2], \frac{1}{2} - \frac{1}{2}x\right)$$

```
> sumtohyper(subs(n=n+2,legendreterm)-
> legendreterm,k);
```

$$(-2\,n + 2\,n\,x - 3 + 3\,x)\,\mathrm{hypergeom}\left(\right.$$
$$\left.[n+2, -n-1], [2], \frac{1}{2} - \frac{1}{2}x\right)$$

Acknowledgement

I thank very much for the warm hospitality which I received at the ETH Zürich, when I stayed there to install my code of Gosper's and Zeilberger's algorithms in the Maple library. My special thanks go to Mike Monagan, who headed the installation, kept interest in all details, and improved the code considerably. Further I thank Tom Koornwinder for his `zeilb` implementation which served as the starting point of ours.

References

[1] R. Pirastu, Algorithms for Indefinite Summation of Rational Functions in Maple, *MapleTech* **2**, pp. 29-38, (1995).

[2] R.W. Gosper, Decision Procedure for Indefinite Hypergeometric Summation, *Proc. Nat. Acad. Sci. U.S.A.* **75**, pp. 40-42, (1978).

[3] W. Koepf, Algorithms for m-fold Hypergeometric Summation. *J. Symbolic Computation* **20**, pp. 399-417, (1995).

[4] D. Zeilberger, A Fast Algorithm for Proving Terminating Hypergeometric Identities, *Discrete Math.* **80**, pp. 207-211, (1990).

[5] D. Zeilberger, The Method of Creative Telescoping, *J. Symbolic Computation* **11**, pp. 195–204, (1991).

[6] D. Zeilberger, A Maple Program for Proving Hypergeometric Identities, *SIGSAM Bulletin* **25**, pp. 4–13, (1991).

[7] Problem 94–2, *SIAM Review* **36**, (1994).

[8] T.H. Koornwinder, On Zeilberger's Algorithm and its q-Analogue: A Rigorous Description, *J. of Comput. and Appl. Math.* **48**, pp. 91–111, (1993).

[9] Y.-K. Man and F.J. Wright, *Fast Polynomial Dispersion Computation and its Application to Indefinite Summation,* Proc. of ISSAC'94 (von zur Gathen ed.), pp. 175-180, (1994).

[10] H.S. Wilf. *Generatingfunctionology,* Academic Press, Boston, (1990).

[11] R.L. Graham, D.E. Knuth and O. Patashnik, *Concrete Mathematics,* Addison-Wesley, Reading, Massachussets, second edition, (1994).

[12] V. Strehl, Binomial Sums and Identities, *MapleTech* **10**, pp. 37-49, (1993).

[13] A.F. Nikiforov, S.K. Suslov and V.B. Uvarov, *Classical Orthogonal Polynomials of a Discrete Variable,* Springer-Verlag, Berlin–Heidelberg–New York, (1991).