**A Variation on Gosper's Algorithm**

*By Andrew Lohr*


**Abstract**

In this paper, we discus an approach to compute indefinite summations. The approach given here differs from the popular Gosper's algorithm for computing hypergeometric sums in two ways. The first not good, in that, unlike Gosper's algorithm, this procedure is not guarenteed to evaluate a summation if a formula for it if one exists. The second good, in that it allows for much broader classes of functions to be summed. For example, we consider the multi basic sequences, which allow for more than just rational functions to be ratios of successive terms, as is the case with hypergeometric sequences.


**Maple packages and data files**

This article is accompanied by the maple package `TEL.txt` which is available with an appendix and several example inputs and outputs at

`http://www.math.rutgers.edu/~ajl213/DrZ/Telescope/readme.html`

**Background**

A sequence $t_n$ is called hypergeometric if there exists polynomials $p(n)$ and $q(n)$ such that

$$\frac{t_{n+1}}{t_n} = \frac{p(n)}{q(n)}.$$

.

Some simple examples of this are (products of) rationals, exponentials, factorials, and binomial coefficients.

Gosper's algorithm [G] gives a technique for "solving" summations of hypergeometric sequences, that is, given a hypergeometric $F(n,k)$, and a sum of the form

$$x_N = \sum_{n=0}^{N-1} F(n),$$

it is able to find a formula for $x_N$ as a constant plus a hypergeometric term, if one exists. For a detailed description of how (and why) Gosper's algorithm works, we refer the reader to Chapter five of the book A=B.[PWZ]

However, many sequences of interest are not hypergeometric, so there is definitely room for further work along this same goal of computing indefinite summations.

**This Article**

Instead of starting with sequences more complicated than hypergeometric, we will start with applying our approach to sequences that are less complicated, and move up from there. Consider rational functions. Suppose that we have some summation

$$x_N = \sum_{n=0}^{N-1} \frac{P(n)}{Q(n)}$$

and we want to say what rational function this is equal to. We know that one exists because we could always of expanded out the summand by rational functions that telescope, notice some cancellation, and then add together what is left over to get a rational expression. In Gosper's algorithm, we are able to tell precisely when some summation is equal to a constant plus a hypergeometric term. However, to pay the price for expanding the possible values that it could sum to, we will have to give up this guarantee of knowing for certain that if our algorithm fails to figure out such a summation, then there is none. Instead, it could be that it only failed for the considered degree of the recurrence, and it may instead find a recurrence that the summation satisfies by simply increasing the order of the recurrence or the degree of the rational functions that are used as coefficients in the recurrence.

First, we try to identify an expression for the limit $L$ of the summation. For our considered summations of a rational summand, we know that the limit will always be rational, and so, by looking at partial fraction decompositions of larger and larger partial summations. Looking at the decomposition, one entry blows up while everything before that entry stays the same, giving us a very good guess that the limit is the part of the decomposition before the entry blowing up. Then, once we know the value of the limit of the summation, we construct the sequence

$$y_N = \sum_{N}^{\infty} \frac{P(n)}{Q(n)},$$

that is,

$$y_N = L - x_N$$

Then, we generate may terms of this sequence and try to identify the result as a hypergeometric sequence. This step in particular allows for a lot of freedom. If instead of hypergeometric, we were looking for descriptions of this sequence as a higher order recurrence, we could just use that information to change the set of linear equations that need to be solved. The other class of functions that we are considering are called multibasic sequences. They are ones where the ratio of successive terms is some fraction of multinomials in different expressions depending on $n$ instead of just polynomials in $n$.

So, for example, to evaluate

$$\sum_{n=0}^{N} -\frac{1}{4}\frac{(2^{2n+2}n^2 + 2n2^{3n} - 2^{4n} + 3n2^{2n} + n2^{n+1} + 4n^2 - 2^{3n} - 72^{2n} - 42^n - 12)}{(n!(2^{2n-2}+1)(2^{2n}+1))}$$

It si found to be

$$e + \frac{(2^N + N + 3 + 2^{2N})}{((2^{2N}+1)N!)}$$

That is, it is Euler's constant plus a multibasic expression in $N$ and $2^N$ of max degree 2.

The drawback of this technique is that of identifying $L$. It works fine if the function is very quickly converging, but it starts to perform poorly as the sum converges more slowly, and is completely worthless when the sum does not have a finite limit. Instead of solving the set of linear equations that we set up, also allow this limit to be an unknown, and solve a resulting non-linear set of equations. Of course this is much slower, and so this techniques is only best used for summations that rapidly converge.

Another broad class of summands to which this can apply are not just when we extend the rato of successive terms to some fixed rational expression of atoms that are hypergeometric,

$$\frac{P(a_1, a_2, \ldots, a_k)}{P(a_1, a_2, \ldots, a_k)}$$

where $a_i$ is something such as $2^n$ or $n!$. Instead, what if we were to allow $a_i$ to be more free in how it depends on $a_i$. For example imagine that we had that successive terms had the ratio $\frac{F_{n+1}}{F_n}$ where $F_n$ for this little example represnets the $n$th Fibonacci number. More generally, the structure about this that is helpful to us is that our new atoms that we are exploting is that they are $C - finite$. This allows us to reduce occurrences of them for larger $n$ to a number of starrting terms equal to the order minus one, possibly times rational expressions in $n$ that come from the recurrence that they satisfy.

**Examples**

Many of the examples here are artificially cooked up, but show the power of this procedure. Any, In general, non-cooked up examples would also yield results. The only problem with that is that the degrees that would be required for for the solution could be very very high. There isn't a way to know before hand how high how high of degrees would need to be considered, so our procedure needs to look for higher and higher degree solutions.

**Using this maple package**

Hopefully by this point, the usefulness of these procedures has been made clear. Though there is more detailed documentation in the maple package itself, here is a breif description of how they are used. To try and figure out a hypergeometric expression for the indefinite sum, allowing degree at most d, and starting at $n = 0$, call $PMG(expression, n, d, 0)$. The procedure will then spit out it's guess for the summation, followed by a line that contains a constant followed by the ratio of consecutive terms satisfied by the sequence. Lastly, it will attempt to return a closed form expression for the summation, which may often not exist.

For The version that is multibasic, instead call $PMGMB$, with the same format for arguments.

**Acknowledgments**

I would like to thank Doron Zeilberger for helping to guide me though this topic.

**References**

[BP] Andrej Bauer, Marko Petkovek, Multibasic and Mixed Hypergeometric Gosper-Type Algorithms, In Journal of Symbolic Computation, Volume 28, Issues 45, 1999, Pages 711-736, ISSN 0747-7171, `https://doi.org/10.1006/jsco.1999.0321.`

[G] Gosper, R. W., Jr., Decision procedure for indefinite hypergeometric summation, Proc. Natl. Acad. Sci. USA 75 (1978), 40-42.

[PWZ] Mario Petkovsek, Herbert Wilf, and Doron Zeilberger, A=B, Taylor & Francis, 1996. `http://www.math.rutgers.edu/~zeilberg/AeqB.pdf`

---

Andrew Lohr, Department of Mathematics, Rutgers University (New Brunswick), Hill Center-Busch Campus, 110 Frelinghuysen Rd., Piscataway, NJ 08854-8019, USA. ajl213 at math dot rutgers dot edu; `http://www.math.rutgers.edu/~ajl213/` .

---

**First Written: August 29, 2017; This version: October 2, 2017.**