# Gosper’s Algorithm (Part 1)

Vanessa Hauri

# Formulas we'll use (helpful to write them down or take a screenshot)

$$z_{n+1} - z_n = t_n \tag{1}$$

$$r(n)y(n+1) - y(n) = 1 \tag{2}$$

$$r(n) = \frac{a(n)c(n+1)}{b(n)c(n)} \tag{3}$$

$$gcd(a(n), b(n+h)) = 1 \tag{4}$$

$$y(n) = \frac{b(n-1)x(n)}{c(n)} \tag{5}$$

$$a(n)x(n+1) - b(n-1)x(n) = c(n) \tag{6}$$

# 1 Introduction

This algorithm allows us to do indefinite hypergeometric sums in simple closed form, or proves the impossibility of it. Let

$$s_n = \sum_{k=0}^{n-1} t_k$$

where $t_k$ is a hypergeometric term not depending on n. Then

$$r(k) = \frac{t_{k+1}}{t_k}$$

is a rational function of k. Goal: express $s_n$ in a closed form.
Note: $s_{n+1} - s_n = t_n$. We want to know if given $t_n$ there exists a hypergeometric term $z_n$ s.t.

$$z_{n+1} - z_n = t_n \tag{1}$$

Note that any such $z_n$ will have the form

$$z_n = z_{n-1} + t_{n-1} = z_{n-2} + t_{n-2} + t_{n-1} = ... = z_0 + \sum_{k=0}^{n-1} t_k = s_n + c$$

where $c = z_0$ is a constant.

**Remark:** Given a hypergeometric term $t_n$, is there a hypergeometric term $z_n$ satisfying $z_{n+1} - z_n = t_n$?
If yes, then $s_n$ can be expressed as a hypergeometric term plus a constant and the algorithm outputs such a term. In that case $t_n$ is called Gosper-summable. If not, then that proves that it has no hypergeometric solution.

# 2 Hypergeometrics to rationals to polynomials

Let $z_n$ be a hypergeometric term satisfying (1). Then

$$z_n = t_n \cdot \frac{1}{\frac{z_{n+1}}{z_n} - 1}$$

is a rational function of n. Let

$$z_n = y(n)t_n$$

where $y(n)$ is a rational function of n. Substituting this for $z_n$ in (1) shows us that $y(n)$ satisfies

$$r(n)y(n+1) - y(n) = 1 \tag{2}$$

where $r(n) = \frac{t_{n+1}}{t_n}$. We have thus reduced the problem of finding hypergeometric solutions of (1) to finding rational solutions of (2).

Assume that we can rewrite

$$r(n) = \frac{a(n)c(n+1)}{b(n)c(n)} \tag{3}$$

where $a(n), b(n), c(n)$ are polynomials in n and it holds that

$$gcd(a(n), b(n+h)) = 1 \tag{4}$$

for all nonnegative integers $h$.

We are looking for a nonzero rational solution of (2) in the form

$$y(n) = \frac{b(n-1)x(n)}{c(n)} \tag{5}$$

where $x(n)$ is an unknown rational function of n. As we substitute (3) and (5) into (2) we see that $x(n)$ satisfies

$$a(n)x(n+1) - b(n-1)x(n) = c(n) \tag{6}$$

**Theorem:** Let $a(n), b(n), c(n)$ be polynomials satisfying $gcd(a(n), b(n+h)) = 1$ for all nonnegative integers $h$. If $x(n)$ is a rational function of n satisfying (6), then $x(n)$ is a polynomial in n.

*Proof.* Outline of the proof: Proof by contradiction. Let $x(n) = f(n)/g(n)$, $f(n)$ and $g(n)$ relatively prime polynomials in n. Rewrite (6) as

$$a(n)f(n+1)g(n) - b(n-1)f(n)g(n+1) = c(n)g(n)g(n+1)$$

$x(n)$ non-polynomial $\implies$ $g(n)$ non-constant polynomial
Let N be st. $gcd(g(n), g(n+N))$ a non-constant polynomial, let $u(n)$ be a non-constant irreducible common divisor of $g(n)$ and $g(n+N)$.
Then $u(n+1)|b(n+N)$ and $u(n+1)|a(n)$. $\implies$ $u(n+1)$ is a non-constant factor of both $a(n)$ and $b(n+N)$ $\implies$ contradicts (4) $\implies$ $x(n)$ polynomial in n. $\square$

If $x(n)$ is a nonzero polynomial solution of (6), then

$$z_n = \frac{b(n-1)x(n)}{c(n)} t_n$$

is a hypergeometric solution of (1) and vice versa.

## Gosper's Algorithm Outline

INPUT: A hypergeometric term $t_n$

OUTPUT: A hypergeometric term $z_n$ satisfying (1) if one exists; $\sum_{k=0}^{n-1} t_k$ otherwise.

1. Form the ration $r(n) = t_{n+1}/t_n$ which is a rational function of $n$.
2. Write $r(n) = \frac{a(n)c(n+1)}{b(n)c(n)}$ where $a(n), b(n), c(n)$ are polynomials satisfying (4).
3. Find a nonzero polynomial solution $x(n)$ of (6) if one exists; otherwise return $\sum_{k=0}^{n-1} t_k$ and stop.
4. Return $\frac{b(n-1)x(n)}{c(n)} t_n$ and stop.

**Example:** Let

$$S_m = \sum_{k=0}^{m} k^2 2^k$$

Let $t_n = n^2 2^n$. Then

$$r(n) = \frac{(n+1)^2 2^{n+1}}{n^2 2^n} = \frac{2(n+1)^2}{n^2}$$

The choices for $a(n), b(n), c(n)$ are obvious, namely $a(n) = 2, b(n) = 1, c(n) = n^2$. It is easy to see that this choice satisfies (3) and (4). Equation (6) thus becomes

$$2x(n+1) - x(n) = n^2$$

Let $x(n) = n(n-4)+6$. Hence,

$$z_n = \frac{1 \cdot x(n) \cdot n^2 2^n}{n^2} = 2^n(n(n-4)+6)$$

which satisfies $z_{n+1} - z_n = t_n$. Finally, $s_m = z_m - z_0 = 2^m(m^2-4m+6)-6$, so the closed form we are looking for is

$$S_m = s_{m+1} = 2^{m+1}(m^2-2m+3)-6$$

# 3 The full algorithm: Step 2

Let $r(n) = f(n)/g(n)$ where $f(n)$ and $g(n)$ are relatively prime polynomials.
If $gcd(f(n), g(n+h)) = 1$ then $a(n) = f(n), b(n) = g(n), c(n) = 1$ gives the desired factorization.
Otherwise let $u(n)$ be a non-constant common factor of $f(n)$ and $g(n+h)$ for some nonnegative integer h. Let $f(n) = \bar{f}(n)u(n)$ and $g(n) = \bar{g}(n)u(n-h)$. Then

$$r(n) = \frac{f(n)}{g(n)} = \frac{\bar{f}(n)u(n)}{\bar{g}(n)u(n-h)}$$

**Defintion:** Given a polynomial $p(x) = a_n x^n + ... + a_0$ of degree $n$ and a polynomial $q(x) = b_m x^m + ... + b_0$ of degree $m$, the resultant is defined as the determinant of their Sylvester Matrix.

Let $R(h)$ denote the resultant of $f(n)$ and $g(n+h)$. Then $R(h)$ is a polynomial in h with the property that $R(\alpha) = 0$ if and only if $gcd(f(n), g(n+\alpha))$ is not a constant polynomial. $\implies$ values of h that violate (4) are the nonnegative integer zeros of $R(h)$.

## Gosper's Algorithm Step 2

2.1. Let $r(n) = Z\frac{f(n)}{g(n)}$ where $f, g$ are monic relatively prime polynomials and Z is a constant.

$R(h) := Resultant_n(f(n), g(n+h))$;

Let $S = \{h_1, h_2, ..., h_N\}$ be the set of nonnegative integer zeros of $R(h)$. ($N \geq 0, 0 \leq h_1 < h_2 < ... < h_N$).

2.2. $p_0(n) := f(n); q_0(n) := g(n)$; for $j = 1, 2, ..., N$ do

$s_j(n) := gcd(p_{j-1}(n), q_{j-1}(n+h_j))$;

$p_j(n) := p_{j-1}(n)/s_j(n)$;

$q_j(n) := q_{j-1}(n)/s_j(n-h_j)$.

$a(n) := Zp_N(n)$;

$b(n) := q_N(n)$;

$c(n) := \prod_{i=1}^{N} \prod_{j=1}^{h_i} s_i(n-j)$.

**Example:** Same example as before, $r(n) = \frac{2(n+1)^2}{n^2}$. Take $Z = 2, f(n) = (n+1)^2, g(n) = n^2$ and note that $f(n)$ and $g(n)$ are relatively prime polynomials. Then

$$R(h) = Resultant_n(f(n), g(n+h)) = (h-1)^4$$

Clearly, the only nonnegative integer zero is $h = 1$. Hence, $s_1(n) = (n+1)^2, p_1(n) = 1, q_i(n) = 1$ which gives us $a(n) = 2, b(n) = 1, c(n) = n^2$.

**Remark:** We can compute directly that the three polynomials produced by this algorithm satisfy condition (3). To show that they also satisfy condition (4), we can note that by definition of $p_j, q_j$, and $s_j$,

$$gcd(p_k(n), q_k(n+h_k)) = gcd(\frac{p_{k-1}(n)}{s_k(n)}, \frac{q_{k-1}(n+h_k)}{s_k(n)}) = 1$$

for all $k$ s.t. $1 \leq k \leq N$.

# Gosper's Algorithm Example

Load the module "gosper":

(Debug) In[198]:=
```
<< "gosper.m"
```

N.B.: Besides GosperSum and GosperFunction, this

package also contains FactorialSimplify (alias FS), and WZ.

Find the ratio r(n):

(Debug) In[199]:=
```
GetRatio[n^2*2^n, n]
```

(Debug) Out[199]=

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

Find y(n) = b(n-1)x(n)/c(n):

(Debug) In[200]:=
```
GosperFunction[2 (1 + n) ^2 / n^2, n]
```

(Debug) Out[200]=

$$\frac{6-4\,n+n^2}{n^2}$$

Multiply this y(n) with t_n to get z_n:

(Debug) In[201]:=
```
n^2*2^n*GosperFunction[2 (1 + n) ^2 / n^2, n]
```

(Debug) Out[201]=

$$2^n\,\left(6-4\,n+n^2\right)$$

Find our Sum by setting S_n=s_(n+1)=z_(n+1)-z_0:

(Debug) In[202]:=
```
2^(n + 1) (6 - 4 (n + 1) + (n + 1) ^2) - 2^0 (6 - 4*0 + 0^2)
```

(Debug) Out[202]=

$$-6+2^{1+n}\,\left(6-4\,(1+n)+(1+n)^2\right)$$

(Debug) In[203]:=
```
Simplify[%]
```

(Debug) Out[203]=

$$-6+2^{1+n}\,\left(3-2\,n+n^2\right)$$

Find the Gosper Sum directly:

(Debug) In[204]:=
```
GosperSum[k^2*2^k, {k, 0, n}]
```

(Debug) Out[204]=

$$-6+2^{1+n}\,\left(3-2\,n+n^2\right)$$

Printed by Wolfram Mathematica Student Edition