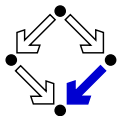


# Special functions, their numerical representations and transcendentals in QCD

M Round  
in collaboration with J Ablinger and J Blümlein.

RISC, Johannes Kepler Universität, Austria.

May, 2017



# Polylogarithms

This talk is an overview of the ongoing problem of special functions in perturbative QCD and their numerical properties.

- 1 Review the basic definitions and properties.
- 2 Goncharov's conjecture (excluding the symbol)
- 3 Numerical evaluation
- 4 PSLQ and transcendentals

# Heavy Flavour Wilson Coefficients

Last week's summation problems were found from 3-loop parton distribution functions (PDFs),

$$\frac{d^2\sigma}{dx dy} = \frac{2\pi\alpha^2}{xyQ^2} [(1 + (1 - y)^2)F_2(x) - y^2F_L(x)],$$

Write this using a product of the (unknown) PDF,  $f_j$ , and a perturbative piece; the Wilson coefficients,

$$F_i(x) = \sum_j \int \frac{dz}{z} C_{i,j} \left(\frac{x}{z}\right) f_j(z),$$

$i \in \{2, L\}$  and  $j$  runs over all (anti-)quarks and the gluon.

Light flavour contributions are known to 3-loops, heavy contributions to 2-loops. This week I will examine numerical evaluation of the real space results which are expressed in terms of polylogarithms.

# Heavy Flavour Wilson Coefficients

Recall that the PDF is introduced as a Mellin convolution,

$$F_i(x) = \sum_j \int \frac{dz}{z} C_{i,j} \left( \frac{x}{z} \right) f_j(z) = \sum_j C_{i,j} * f_j.$$

Thus by taking the Mellin transform,

$$\mathcal{M}[g](N) = \int_0^1 dx x^{N-1} g(x) \quad N \in \mathbb{Z}^+$$

things simplify greatly. In Mellin space Feynman diagrams contributing to the Wilson coefficients  $C_{i,j}$  become rational in the Mellin parameter  $N$  and definite sums involving  $N$ . This week we need position space, the inverse transform of results must be computed.

# Mellin transforms

Last week sums were considered in detail and the output was in terms of various  $\Pi\Sigma$ -extensions to a base field. These  $\Pi\Sigma$ -extensions are types of harmonic sums and their generalisations. Recall,

$$S_{\vec{k}}(n) = \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} \cdots \sum_{i_k=1}^{i_{k-1}} \prod_{j=1}^k \frac{[\text{sign}(k_j)]^{i_j}}{i_j^{|k_j|}}.$$

Computing the inverse Mellin transform is another topic today I want to discuss the types of function one finds and what can be done with them.

# Polylogarithms

## Parameter integrals

To compute the Wilson coefficients one must compute Feynman diagrams (with the additional feature of an operator insertion). The standard practice is to introduce Feynman parameters,

$$\frac{1}{A_1 \cdots A_m} = \Gamma(m) \int_0^1 dx_1 \cdots dx_m \frac{\delta(x_1 + \cdots + x_m - 1)}{(x_1 A_1 + \cdots + x_m A_m)^m}$$

One may apply the  $\delta$ -function which introduces a step function  $\Theta$ ,

$$x_1 = 1 - x_2 - \cdots - x_m \in (0, 1),$$
$$\int_0^1 dx_2 \cdots dx_m \frac{\Theta(1 - x_2 - \cdots - x_m)}{([1 - x_2 - \cdots - x_m]A_1 + \cdots + x_m A_m)^m}$$

# Polylogarithms

## Letters

This can be re-arranged via some changes of variables into a sum of nested integrals,

$$\int_0^1 dt_1 f(t_1) \int_0^{t_1} f(t_2) dt_2 \cdots \int_0^{t_{n-1}} dt_n f(t_n).$$

What is not clear is how to choose the  $f_i$ . Essentially the most basic objects of interest are,

$$\frac{1}{x - a_i},$$

The  $a_i$  are referred to as letters and intuitively represent dimensionless combinations of scales arising in the problem. Iterated integrals over letters  $f_i$  are known as polylogarithms with various prefixes.

# Polylogarithms

## Example

This certainly does not cover all possible Feynman diagrams — there are parameter integrals that can not be cast in terms of nested integrals over  $f_i$ . However one can obtain a lot of diagrams this way and these objects are already very general. While elementary this is sufficiently important to warrant an example. Consider,

$$\begin{aligned}\frac{1}{ABC} &= \int_0^1 dx_i \frac{\delta(1 - x_1 - x_2 - x_3)}{[x_1 A + x_2 B + x_3 C]^3}, \\ &= \int_0^1 dx_1 \int_0^{1-x_1} dx_2 \frac{1}{[x_1 A + x_2 B + (1 - x_1 - x_2) C]^3}, \\ &= - \int_0^1 dx_1 \int_0^{x_1} dx_2 \frac{1}{[x_1 A + x_2 B + (x_2 - x_1) C]^3}, \\ &+ \int_0^1 dx_1 \int_0^1 dx_2 \frac{1}{[x_1 A + x_2 B + (x_2 - x_1) C]^3}\end{aligned}$$



# Polylogarithms

## Example

- Next one would like to perform these integrals. Like last week, some part will be recognizable in terms of elementary functions and known special functions like Gauß's hypergeometric function. What remains are new iterated integrals that occur in field theory.
- After momentum integration the  $A$ ,  $B$  and  $C$  will give scales that appear as parameters in these special functions.
- From this angle a Feynman diagram can be viewed as the problem of expressing the input in terms of known special functions and various iterated integrals. Simplifying, understanding and evaluating those iterated integrals is the topic of this talk.

# Polylogarithms

## Literature

The 'earliest' occurrence of a polylogarithm both in mathematics and particle physics is usually the dilogarithm,

$$\operatorname{Li}_2(x) = - \int_0^x dt \frac{\log(1-t)}{t} = - \int_0^x \frac{dt_1}{t_1} \int_0^{t_1} \frac{dt_2}{t_2 - 1}.$$

the first integral is for  $z \in \mathbb{C}$  the second for  $|z| < 1$ . Notice that one might be tempted to define the dilogarithm as,

$$\int_0^x \frac{dt_1}{t_1} \int_0^{t_1} \frac{dt_2}{1-t_2}.$$

Much is known about the dilogarithm. Below are just 2 example relations,

$$\operatorname{Li}_2(1) = \zeta(2), \quad \operatorname{Li}_2(z) + \operatorname{Li}_2(-z) = \frac{1}{2} \operatorname{Li}_2(z^2).$$

# Polylogarithms

## Harmonic polylogarithms

Vermaseren and Remiddi defined the harmonic polylogarithms (HPLs),

$$f_1(x) = \frac{1}{1-x}, \quad f_0(x) = \frac{1}{x}, \quad f_{-1}(x) = \frac{1}{x+1},$$

which *violates* the convention choice made earlier. With these the harmonic polylogarithms are now defined,

$$H(\vec{a}; x) := \int_0^x dt f_{a_1}(t) H(\vec{a}_{i>1}; t), \quad H(\emptyset; x) = 1, \quad H(\vec{0}; x) = \frac{1}{w!} \log^w x.$$

The length of  $\vec{a} =: w$  is the weight of the polylogarithm. For example,

$$H(0, 1; x) = \int_0^x dt \frac{H(1; t)}{t} = \int_0^x dt \frac{1}{t_1} \int_0^{t_1} dt_2 \frac{1}{1-t_2} = \text{Li}_2(x).$$

# Algebraic Relations

## Properties

Harmonic polylogarithms enjoy several properties. The shuffle product  $A \sqcup B$  is the order preserving interlacing of two sets,

$$\{a, b\} \sqcup \{c\} = \{\{a, b, c\}, \{a, c, b\}, \{c, a, b\}\}$$

then the harmonic polylogarithms satisfy,

$$H(\vec{a}; x)H(\vec{b}; x) = \sum_{\vec{c} \in \vec{a} \sqcup \vec{b}} H(\vec{c}; x),$$

which are referred to as shuffle identities. Which is much clearer with an example,

$$\begin{aligned} H(0, 1; x)H(1; x) &= H(0, 1, 1; x) + H(0, 1, 1; x) + H(1, 0, 1; x), \\ &= 2H(0, 1, 1; x) + H(1, 0, 1; x). \end{aligned}$$

# Algebraic Relations

## Properties

A second class of identities can be derived from integration by parts. One obtains the following general formula,

$$\begin{aligned} H(\vec{a}; x) &= H(a_1; x)H(\vec{a}_{i>1}; x) - \int_0^x dt f_{a_2}(t)H(a_1; t)H(\vec{a}_{i>2}; t), \\ &= H(a_1; x)H(\vec{a}_{i>1}; x) - H(a_2, a_1; x)H(\vec{a}_{i>2}; x) + \cdots \\ &\quad + (-1)^{w+1}H(a_w, \dots, a_1; x). \end{aligned}$$

Again a simple example is,

$$\begin{aligned} H(1, 1; x) &= H(1; x)H(1; x) - H(1, 1; x), \\ \Rightarrow H(1, 1; x) &= \frac{1}{2}H(1; x)^2. \end{aligned}$$

# Algebraic Relations

## Basis

Having such relations the natural task is to eliminate all algebraic relations and form a basis. (An algebraic relation is one involving positive integer powers of harmonic polylogarithms all evaluated at  $x$  and real coefficients.) There are  $3^2 = 9$  weight 2 HPLs  $\frac{3 \times 2}{2}$  shuffle relations and 3 IBP relations so a weight 2 basis is,

$$H(0, 1; x), \quad H(-1, 1; x), \quad H(-1, 0; x).$$

Some of the relations for the other weight 2 HPLs are,

$$\begin{aligned} H(1, -1; x) &= H(-1; x)H(1; x) - H(-1, 1; x), \\ H(0, 0; x) &= \frac{1}{2}H(0; x)^2. \end{aligned}$$

For a review see for example the lecture notes by [ABLINGER & BLÜMLEIN '13](#) and references therein.

# Generalisations

## Goncharov polylogarithm

More complicated Feynman diagrams lead one to observing complex letters which suggests to define the Goncharov polylogarithm,

$$G(\vec{a}; x) := \int_0^x \frac{dt}{t - a_1} G(\vec{a}_{i>1}; t), \quad G(\emptyset; x) = 1, \quad G(\vec{a}; 0) = 0,$$

for  $a_i \in \mathbb{C}$ . (Depending on  $x \in \mathbb{C}$  it may be that there is a pole on the contour of integration this can be resolved but I will not discuss the topic.) Beyond the shuffle and integration by parts identities the Goncharov polylogarithm also has a scaling property,

$$G(\vec{a}; x) = G(\alpha \vec{a}; \alpha x),$$

for  $\alpha \in \mathbb{C}$ . Note however that there are minus signs when comparing an HPL to a Goncharov polylogarithm due to differing conventions,

$$G(0, 1; x) = -H(0, 1; x).$$

# Generalisations

## Hölder convolution

Goncharov polylogarithms also obey the Hölder convolution BORWEIN '01,

$$G(\vec{a}; 1) = \sum_{j=0}^w (-1)^j G(1 - a_j, 1 - a_{j-1}, \dots, 1 - a_1; 1 - p) G(a_{j+1}, \dots, a_w; p)$$

for  $p \in \mathbb{C}$  arbitrary and  $a_1 \neq 1$  and  $a_w \neq 0$ . Again an example cuts through the notation,

$$G(a, b; 1) = G(\emptyset; 1 - p) G(a, b; p) - G(1 - a; 1 - p) G(b; p) \\ + G(1 - b, 1 - a; 1 - p) G(\emptyset; p).$$

Relations of this type, depending on  $p$  are analytic; the argument changes and therefore are independent of how the basis was set-up. Thus one concludes that Goncharov polylogarithms satisfy (potentially) far more relations than the harmonic polylogarithm subset.



# Generalisations

## Multiple polylogarithms

There is yet another object of interest, the multiple polylogarithm, which is equivalent to the Goncharov polylogarithm but highlights some properties better.

$$\mathrm{Li}_{\vec{m}}(\vec{x}) = (-1)^w G\left(\vec{0}_{m_1-1}, \frac{1}{x_1}, \vec{0}_{m_2-1}, \frac{1}{x_1 x_2}, \dots, \vec{0}_{m_w-1}, \frac{1}{\prod_{i=1}^w x_i}; 1\right)$$

and now  $w$  is the length of  $x$  or equivalently  $\vec{m}$ . Setting all arguments equal to one,  $\vec{x} = \vec{1}$ , gives the so-called multiple- $\zeta$  values,

$$\mathrm{Li}_{\vec{m}}(\vec{1}) = \zeta_{\vec{m}}.$$

In one dimension,  $\vec{m} \in \mathbb{N}$ , this further reduces to the usual Riemann  $\zeta$ -function at natural arguments.

# Generalisations

## Stuffle relations

Multiple polylogarithms obey a stuffle relation which is quite involved to understand and I will omit. In short, rather than using a shuffle product to combine to lists of indices one uses a stuffle product. The relation is of the form,

$$\mathrm{Li}_{\vec{m}}(\vec{x})\mathrm{Li}_{\vec{n}}(\vec{y}) = \sum_i \mathrm{Li}_{\vec{r}_i}(\vec{s}_i),$$

the  $\vec{r}_i$  and  $\vec{s}_i$  come out of the stuffle product. An example for a little more clarity,

$$\mathrm{Li}_m(x)\mathrm{Li}_n(y) = \mathrm{Li}_{m,n}(x,y) + \mathrm{Li}_{n,m}(y,x) + \mathrm{Li}_{m+n}(xy)$$

One can of course translate this into the Goncharov notation. Stuffle relations are in addition to shuffle relations — not every GPL is a MPL!

# Goncharov Conjecture

## Basis redux

For a set of Goncharov polylogarithms there is the previous basis calculation where shuffle and integration by parts relations are removed. However, there are also the stuffle relations and the Hölder convolution plus the potential for unknown relations. This leads us into Goncharov's conjecture.

## Conjecture

Any multiple polylogarithm beyond weight 1 may be expressed in terms of multiple polylogarithms without an index equal to 1.

This conjecture allows one to write down a handful of required functions. For example at weight 5 the conjecture implies only,

$$\text{Li}_5(x), \quad \text{Li}_{3,2}(x, y), \quad \text{Li}_4(x), \quad \text{Li}_{2,2}(x, y), \quad \text{Li}_2(x), \quad \log(x).$$

are needed. While  $\text{Li}_{1,2}(x, y)$  is not needed for example.

# Goncharov Conjecture

## Computation

Let us perform this computation at weight 4 using the Mathematica package HarmonicSums. The start point is all possible weight 4 Goncharov polylogarithms; essentially one needs all possible ways of writing 4 different indices, 3 different indices, 2 different and finally a single index. After removing all shuffle and integration by parts relations there are 15 such polylogarithms which include,

$$G(a, b, c, d; 1), \quad G(a, a, b, b; 1), \quad G(a, b, b, b; 1).$$

Now it is possible to use two Hölder convolutions to 'insert a zero' into any polylogarithm without a zero index.

# Goncharov Conjecture

## Computation

Let us tackle a specific weight 2 example however the method is general.  
Given  $G(a, b; x)$   $a, b \neq 0$  normalise the last index to 1,

$$G(a, b; x) = G\left(\frac{a}{b}, 1; \frac{x}{b}\right).$$

Compute a related Hölder convolution with  $p = \frac{x}{b}$ ,

$$\begin{aligned} G\left(\frac{a}{b}, 1; 1\right) &= -G\left(1; \frac{x}{b}\right) G\left(1 - \frac{a}{b}, 1 - \frac{x}{b}\right) + G\left(0, 1 - \frac{a}{b}, 1 - \frac{x}{b}\right) \\ &\quad + G\left(\frac{a}{b}, 1; \frac{x}{b}\right) \end{aligned}$$

Re-arranged for  $G\left(\frac{a}{b}, 1; \frac{x}{b}\right)$  there is still a  $w = 2$  polylogarithm without a zero index. Compute a second Hölder convolution now with  $p = 0$ ,

$$G\left(\frac{a}{b}, 1; 1\right) = G\left(1 - \frac{a}{b}; 1\right) + G\left(0, 1 - \frac{a}{b}; 1\right).$$

# Goncharov Conjecture

## Computation

Combining the two Hölder convolution results allows one to write out that,

$$G(a, b; x) = G\left(1 - \frac{a}{b}; 1\right) + G\left(0, 1 - \frac{a}{b}; 1\right) + G\left(1; \frac{x}{b}\right) G\left(1 - \frac{a}{b}, 1 - \frac{x}{b}\right) - G\left(0, 1 - \frac{a}{b}, 1 - \frac{x}{b}\right).$$

Everything on the right-hand side is of lower weight or contains a zero. This procedure of computing two suitable Hölder convolutions in exact analogy to here always allows one to insert a zero into a non-zero index vector. Performing this to the weight 4 list of Goncharov polylogarithms and tidying up gets one down to the following list,

$$G(0, a, b, c; 1), \quad G(0, 0, a, b; 1), \quad G(0, 0, 0, a; 1), \quad G(0, a, 0, b; 1).$$

# Goncharov Conjecture

## Computation

Translating to the multiple polylogarithm notation the list contains,

$$\text{Li}_{2,1,1}, \quad \text{Li}_{3,1}, \quad \text{Li}_4, \quad \text{Li}_{2,2}.$$

Now that we are in the multiple polylogarithm language one can apply the stuffle relations. Elimination of stuffle relations is not so neatly automated in HarmonicSums (yet!) so we do it by hand. Compute that,

$$\text{Li}_3(x)\text{Li}_1(y) = \text{Li}_4(xy) + \text{Li}_{1,3}(y, x) + \text{Li}_{3,1}(x, y).$$

Now compute two shuffle relations,

$$\text{Li}_1(x)\text{Li}_3(xy) = \text{Li}_{1,3}(y, x) + \text{Li}_{2,2}(y, x) + \text{Li}_{3,1}(y, x) + \text{Li}_{3,1}(xy, \frac{1}{x}),$$

$$\text{Li}_2(y)\text{Li}_2(xy) = \text{Li}_{2,2}(y, x) + \text{Li}_{2,2}(xy, \frac{1}{x}) + 2\text{Li}_{3,1}(y, x) + 2\text{Li}_{3,1}(xy, \frac{1}{x}).$$

# Goncharov Conjecture

## Computation

One can solve these 3 relations to eliminate  $\text{Li}_{3,1}$  in favour of  $\text{Li}_4$  and  $\text{Li}_{2,2}$  reducing us to the set,

$$\text{Li}_{2,1,1}, \quad \text{Li}_4, \quad \text{Li}_{2,2}.$$

$\text{Li}_{2,1,1}$  or equivalently  $G(0, a, b, c; 1)$  remains. I will not remove it because it is a problem outside computer algebra; there is no known algorithm to return an expression in terms of appropriate multiple polylogarithms. The result is known though FRELLESVIG '16.



# Goncharov Conjecture

## Computation

More generally this represents the current situation with regards to Goncharov's conjecture. It is possible, with existing algorithms to,

- return a small number of integrals that must be computed using analysis techniques or using some under appreciated combinations of known results.
- Explicitly verify the conjecture up to weight 4.
- Further relations, that help the conjecture may be found with numerics or intuition.

# Numerics

## Series

Having calculated Feynman diagrams and settled on a representation in terms of special functions the next issue is how to evaluate such expressions.

- A simple approach would be direct numerical integration however a better method is to find a rapidly converging series representation.

For  $x \in (0, 1)$  one may repeatedly apply a Maclaurin expansion to expand letters,

$$\frac{1}{x-a} = - \sum_{r=0}^{\infty} \frac{x^r}{a^{r+1}},$$

then integrate term by term to derive a sum for the whole polylogarithm. (Polylogarithms whose trailing index is zero do not directly admit such an expansion because of the  $\log(x)$  which can not be expanded so one must extract this piece by shuffle products.)

# Numerics

## Series

Such a series will converge provided  $\forall i |x| < |a_i|$ . For HPLs the indices are  $\{1, 0, -1\}$  so provided  $|x| \ll 1$  the series is useful. For larger arguments one can use a variable transformation. In particular if,

$$x = \frac{1-t}{1+t} \Rightarrow x \in (\sqrt{2}-1, 1) \Leftrightarrow t \in (0, \sqrt{2}-1).$$

Additionally by writing out an HPL,

$$H\left(\vec{a}; \frac{1-t}{1+t}\right) = \int_0^{\frac{1-t}{1+t}} d\tau f_{a_1}(\tau) H(\vec{a}_{i>1}; \tau),$$

then making successive variable transformations one can express the input in terms of HPLs at  $t$ . For example,

$$H\left(1; \frac{1-t}{1+t}\right) = -H(-1; 1) + H(-1; t) - H(0; t)$$

Everything on the left is at small argument so can be evaluated by a series representation.

# Numerics

## Bernoulli speed-up

For HPLs of arbitrary weight the above method suffices; series expansion for small argument then variable transformations to map from elsewhere on the complex plane to small argument HPLs.

- There is an additional speed-up option, the Bernoulli expansion, that helps with efficiency.
- At the innermost integral of any polylogarithm is a logarithm. The series expansion derived always contains an expansion of that logarithm.
- Better then is to write the expansion out in terms of logarithmic quantities.

# Numerics

## Bernoulli speed-up

Recall the dilogarithm definition,

$$\begin{aligned}\operatorname{Li}_2(x) &= -\int_0^x dt \frac{\log(1-t)}{t} = \int_0^{-\log(1-x)} du \frac{u}{e^u - 1}, \\ &= \sum_{i=0}^{\infty} \frac{B_i}{(i+1)!} (-\log(1-x))^{i+1}.\end{aligned}$$

This new expansion, using the Bernoulli numbers  $B_i$ , has much improved convergence because ‘the logarithm is not expanded’.

# Numerics

## Bernoulli speed-up

This leads to whether one can apply the Bernoulli speed-up to HPLs? The answer is yes but there is a significant complication from the branch cut structure. Consider  $H(-1, 1; x)$  which contains branch cuts corresponding to both logarithms  $\log(1 - x)$  and  $\log(1 + x)$ . To obtain a good series expansion one must separate out the polylogarithm into the two branch pieces,

$$H(-1, 1; x) = \int_0^x dt \frac{H(1; t) - H(1; -1)}{t + 1} + \int_0^x dt \frac{H(1; -1)}{t + 1}$$

Separating the branch cut structure leads to a better behaved, and so faster, expansion.

# Numerics

## Bernoulli speed-up

One may recursively compute a separation of the HPL into two pieces with better analytic structure which can then be expanded using an appropriate logarithm. Avoiding a full discussion of this technicality one can write VOLLINGA '04,

$$H(\vec{a}; x) = \sum_{i=0}^{\infty} \frac{C_{\vec{a}}(i)}{(i+1)!} (-\log(1-x))^{i+1}$$

where the  $C$  is quite a lot of work to compute,

$$C_{1, \vec{a}}(i) = \begin{cases} 0 & i = 0 \\ C_{\vec{a}}(i-1), & i > 0 \end{cases}, \quad C_{a_1+1, \vec{a}}(i) = \sum_{j=0}^i \binom{i}{j} \frac{B_{i-j}}{j+1} C_{a_1, \vec{a}}(j).$$

# Numerics

## Bernoulli speed-up

Then for a single index one uses,

$$C_1(i) = \delta_{0,i}, \quad C_{n+1}(i) = \sum_{j=0}^i \binom{i}{j} \frac{B_{i-j}}{j+1} C_n(j).$$

There is clearer treatment in GEHRMANN '01 where one considers the analytic structure of the HPLs and proceeds accordingly but it would be a technical summary of known ideas so the formulae are better here.

For a choice of basis up to weight 8 a FORTRAN library of the HPLs on  $(0, 1)$  is available.



# Numerics

## Goncharov Polylogarithm

Numerical evaluation of the Goncharov polylogarithm is significantly harder. Recall series convergence is poor when the argument is close to an index. With arbitrary complex indices allowed in general one can not always arrange to write a polylogarithm in a series that converges quickly. For example a variable transformation will generate new letters,

$$G\left(2; \frac{1-t}{1+t}\right) = G(-1; t) - G\left(-\frac{1}{3}; t\right) - G(0; 1) + G(0; 2).$$

The presence of  $-\frac{1}{3}$  means variable transformations are not a general solution. Alternatively, one could return to the symbol and Goncharov's conjecture. Up to weight 4 it is possible to write out all Goncharov polylogarithms in terms of just a handful of functions of which only  $\text{Li}_{2,2}$  is not a well-understood function. For an implementation see FRELLESVIG '16.

# Numerics

## GiNaC Recipe

In GiNaC there is another approach which can be slow but completely general.

- Recall our task is to write a given Goncharov polylogarithm in terms of polylogarithms with indices larger than the argument in magnitude.
- A sequence of variable transformations and integration by parts identities applied allows to write a series representation that converges.

A convergent series representation is delivered — it may converge slowly.

# Numerics

## GiNaC Recipe

A good implementation for experimental Numerics would be a tabulation to fixed precision. If the series has no good convergence properties one can not be sure of how many terms to tabulate. The dangerous scenario is an index arbitrarily close to the argument,

$$a_i \simeq x.$$

# Numerics

## GiNaC Recipe

To speed-up convergence one can use the Hölder convolution. Recall  $G(\vec{a}; x)$  converges poorly if there exists  $a_j \simeq x$  with  $|x| < |a_j|$ . Normalise the polylogarithm of interest to have argument one then use the Hölder convolution with  $p = \frac{1}{n}$  to write,

$$\begin{aligned} G(\vec{b}; 1) &= \sum_{j=0}^w (-1)^j G(1 - b_j, 1 - b_{j-1}, \dots, 1 - b_1; \frac{1}{n}) G(b_{j+1}, \dots, b_w; \frac{1}{n}), \\ &= \sum_{j=0}^w (-1)^j G(n - nb_j, n - nb_{j-1}, \dots, n - nb_1; 1) \\ &\quad \times G(nb_{j+1}, \dots, nb_w; 1) \end{aligned}$$

with  $\vec{b} = \vec{a}/x$ . By assumption this converges so,

$$|b_j| > 1.$$

# Numerics

## GiNaC Recipe

After applying the Hölder convolution the convergence would be better if,

$$|nb_j| > |b_j| > 1 \Rightarrow n > 1, \quad |n(1 - b_j)| = |n||1 - b_j| > |1 - b_j| > 0\#.$$

Thus one can not ensure that the polylogarithms in the Hölder convolution will converge. One must repeat the algorithm of transformations to express a polylogarithm in terms of convergent objects. Those objects that do converge, which includes the  $j = 0$  term,

$$G(nb_1, \dots, nb_w; 1).$$

have improved convergence. Ultimately this procedure terminates and the resulting expression has improved convergence but GiNaC trades whether it is worth trying to improve convergence!

# Cyclotomic polylogarithms

## Formalism

Our motivation for using the Goncharov polylogarithm is mathematical, physically quantities are real so it should be possible to find real representations of (at least some of) the Goncharov polylogarithms that occur in Feynman diagram calculations. These are the cyclotomic polylogarithms. The  $n^{\text{th}}$  cyclotomic polynomial is the unique factor free polynomial which divides  $x^n - 1$  but no  $x^k - 1$  for  $k < n$ .

$$\Phi_1(x) = x - 1,$$

$$\Phi_3(x) = x^2 + x + 1,$$

$$\Phi_5(x) = x^4 + x^3 + x^2 + x + 1,$$

$$\Phi_2(x) = x + 1,$$

$$\Phi_4(x) = x^2 + 1,$$

$$\Phi_6(x) = x^2 - x - 1,$$

They are available in Mathematica using `Cyclotomic[n,x]`.

# Cyclotomic polylogarithms

## Formalism

A cyclotomic polylogarithm is formed using letters,

$$f_k^l(x) = \frac{x^l}{\Phi_k(x)}.$$

Any polynomial of degree  $n$  has  $n$  roots in the complex plane so in principle one can factorise the cyclotomic polylogarithms and perform a partial fraction decomposition,

$$\begin{aligned}\Phi_6(x) &= (-(-1)^{\frac{1}{3}} + x)((-1)^{\frac{2}{3}} + x) \\ \Rightarrow \frac{1}{\Phi_6(x)} &= \frac{(-1)^{2/3}}{(1 + \sqrt[3]{-1})(x + (-1)^{2/3})} + \frac{(-1)^{2/3}}{(1 + \sqrt[3]{-1})(\sqrt[3]{-1} - x)}.\end{aligned}$$

This represents a general property; the cyclotomic polynomials factor into roots of unity. A cyclotomic polylogarithm is a Goncharov polylogarithm using only roots of unity and 0 for its indices.

# Cyclotomic polylogarithms

## Formalism

Issues concerning Goncharov polylogarithms are still to be resolved. Cyclotomic polylogarithms are far better behaved and understood. Immediately the convergence condition is satisfied for  $x \in (0, 1)$ ,

$$|a_i| = 1 \Rightarrow |x_i| \leq |a_i|,$$

with a roughly constant convergence property for any cyclotomic polylogarithm. This means one can tabulate series and provide an efficient implementation. One is available to weight 4 and for cyclotomy 6.



# Cyclotomic polylogarithms

## Constants

For large arguments one can use the same map as in the HPL case,

$$x = \frac{1-t}{1+t}.$$

Like before, constants are generated that refer to cyclotomic polynomials evaluated at 1. In the HPL case these numbers are well understood. For the Goncharov case little is known about these numbers and it is worth trying! This helps better understand possibly unknown relations. Pick a particular set of cyclotomic letters,

$$x, \quad f_1^0, \quad f_2^0, \quad f_3^1, \quad f_3^1, \quad f_4^0, \quad f_4^1, \quad f_6^0, \quad f_6^1.$$

Next generate all possible cyclotomic polylogarithms evaluated at one. (These are essentially Goncharov polylogarithms evaluated at up to 6<sup>th</sup> roots of unity.)

# Cyclotomic polylogarithms

## Constants

Eliminate all shuffle, stuffle and integration by parts relations from these numbers meaning we have reduced to a basis. Additionally there is another set of relations, the distributive relations, which occur for polylogs at 1 with roots of unity indices (ZHAO '08.) Up to weight 4 one obtains,

weight	All	Basis
1	$1^9 = 9$	4
2	$2^9 = 81$	2
3	$3^9 = 729$	7
4	$4^9 = 6,561$	18

It remains to ask, could there be undiscovered relations (ZHAO '08)?

## PSLQ

We have seen many algebraic relations but also that there is still more to the story. To fully solve aspects of this issue one can perform numeric studies using the PSLQ algorithm.

### PSLQ

Given a vector  $\vec{v}$  construct when possible a  $\vec{u} \in \mathbb{Z}^n$  with  $|\vec{u}| \leq d$  such that  $\vec{u} \cdot \vec{v} = 0$ .

A simple example would be to use Mathematica's built-in function `FindIntegerNullVector` on  $(\pi^2, \zeta(2))$ . Mathematica returns,

$$\{1, -6\}.$$

Using GiNaC's evaluation tool one can evaluate the cyclotomic constants to some large number of digits, such as 10,000, then look for integer relations. PSLQ implementations allow one to control  $d$  which stops one finding relations with arbitrarily large coefficients.

# PSLQ

## Results

Broadhurst studied a problem of cyclotomic constants — cyclotomic polylogs at 1 — up to weight 3 and gave a set of numbers,

$$\pi, \zeta(2), \zeta(3), \log(2), \log(3), \operatorname{Cl}_2\left(\frac{\pi}{3}\right) := \Im \operatorname{Li}_2(e^{i\pi/3}), \operatorname{Li}_2\left(\frac{1}{4}\right),$$

plus the real and imaginary parts of,

$$\operatorname{Li}_3\left(\frac{e^{i\pi/6}}{2}\right), \quad \operatorname{Li}_3\left(\frac{i}{\sqrt{3}}\right)$$

Recall that  $\pi$  and  $\zeta(2)$  obey no PSLQ relation because  $\zeta(2) \propto \pi^2$ .

Broadhurst and Bailey produced a FORTRAN PSLQ implementation (that is much better than Mathematica's). In principle one can write out vectors of various numbers and see whether for some  $d$  there are any relations amongst the numbers.

# PSLQ

## Basis construction

The exercise we performed was a full basis construction. Namely one computes all relations at weight 1 amongst the cyclotomic constants and the numbers Broadhurst gave. Then for weight 2, one computes all weight 2 numbers possible from a minimal set of the weight 1 numbers. Again one performs a PSLQ search to eliminate all relations between these elements and the weight 2 cyclotomic constants. Continuing to weight 4 the results are,

Weight	All	Basis	PSLQ
1	$1^9 = 9$	4	4
2	$2^9 = 81$	2	2
3	$3^9 = 729$	7	5
4	$4^9 = 6,561$	18	10

for the cyclotomic polylogarithms at 1.

# PSLQ

## Basis construction

A second exercise concerns the constants required to tabulate a numerical implementation of the cyclotomic polylogarithms to weight 4 with cyclotomy 6. These numbers originate from the,

$$x = \frac{1 - t}{1 + t}$$

transform and there are slightly more of them. They fully include the previous set. The results are,

Weight	Basis	New Numbers
1	5	1
2	8	3
3	41	14
4	185	?

The quantities of interest are in the form of real and imaginary parts; taking the numbers themselves there are only 155.

# PSLQ

## Relations

Most of these relations are checked to 10,000 digits while some later ones are new and are not verified to extremely high precision yet. Some example relations are ones missed by the basis construction. For example,

$$\frac{5}{4} \Im G(e^{-i\frac{\pi}{6}}; 1) = \Im G(e^{-i\frac{\pi}{3}}; 1) = -\frac{\pi}{3}.$$

This is an analytic relation not seen by the shuffle, stuffle, distributive and integration by parts identities. Or,

$$\Re G(e^{-i\frac{\pi}{6}}, e^{i\frac{5\pi}{6}}; 1) = -\frac{11}{288} \pi^2 - \frac{1}{2} \left[ \Re G(e^{-i\frac{\pi}{6}}; 1) \right]^2 - 4 \Re G(e^{-i\frac{5\pi}{6}}, e^{i\frac{\pi}{3}}; 1).$$

Proving these relations from integral representations may be quite straightforward but one must know where to look.

# Summary

- 1 Today I have review the ongoing story of special functions in particle physics.
- 2 There are many relations and a lot known about polylogarithms but open issues.
- 3 Fast numerical evaluation remains an issue for Goncharov polylogarithms.
- 4 There are still undiscovered relations amongst polylogarithms and their particular values.