DESY 13–064
DO–TH 17/12
September 2018

arXiv:1809.07084 [hep-ph]

Numerical Implementation of Harmonic Polylogarithms to Weight $\mathsf{w} = 8$

J. Ablinger[a], J. Blümlein[b], M. Round[a,b], and C. Schneider[a],

[a] *Research Institute for Symbolic Computation (RISC), Johannes Kepler University, Altenbergerstraße 69, A–4040 Linz, Austria*

[b] *Deutsches Elektronen–Synchrotron, DESY, Platanenallee 6, D-15738 Zeuthen, Germany*

Abstract

We present the `FORTRAN`-code `HPOLY.f` for the numerical calculation of harmonic polylogarithms up to $\mathsf{w} = 8$ at an absolute accuracy of $\sim 10^{-15}$ or better. Using algebraic and argument relations the numerical representation can be limited to the range $x \in [0, \sqrt{2}-1]$. We provide replacement files to map all harmonic polylogarithms to a basis and the usual range of arguments $x \in \,]-\infty, +\infty[$ to the above interval analytically. We also briefly comment on a numerical implementation of real valued cyclotomic harmonic polylogarithms.

Program Summary

Title of program: `HPOLY.f`
Version: 1.00 September 2018
Catalogue number:
Program obtainable from: `https://www-zeuthen.desy.de/~blumlein/`
e-mail: `Johannes.Bluemlein@desy.de`
Licensing provisions: CC by NC 3.0
Programming Language: Fortran
Supplementary Material: none
Journal Reference of previous version: none
Does the new version supersede the previous version?: no
Reasons for the new version: none
Summary of revisions: no revisions
Solution method: Bernoulli-improvement of series expansions, Ref. [1].
Additional comments including Restrictions and Unusual features: none
Computers: all
Operating system: all
Processor used: Intel(R) Core(TM) i7-6600U CPU @ 2.60GHz. All run times given refer to this processor.
Memory required for the executable: 57 kB
No. of bits per word: 64
No. of lines in distributed program: 1291
Other programs called: none.
External files needed: (a) Input data: B51.m, B52.m, B53.m, B54.m, B55.m, B61.m, B62.m, B63.m, B64.m, B65.m, B66.m, B71.m, B72.m, B73.m, B74.m, B75.m, B76.m, B77.m, B81.m, B82.m, B83.m, B84.m, B85.m, B86.m, B87.m, B88.m, TTT2.m, TTT3.m, TTT4.m, TTT5.m, TTT6.m, TTT7.m, TTT8.m
(b) Bases: LIST1.m, LIST2.m, LIST3.m, LIST4.m, LIST5.m, LIST6.m, LIST7.m, LIST8.m, LIST.m
(c) `Mathematica` replacement files: hrel, LISTMTX.m, LISTOMX.m, LISTOOX.m,
Keywords: harmonic polylogarithms, to weight $\mathtt{w} = 8$, numerical implementation, Nielsen integrals, classical polylogarithms.
Nature of physical problem: Harmonic polylogarithms form key building blocks in the analytic calculation of many multi-loop and multi-leg calculations in high energy physics and other disciplines of modern physics. Their numerical representation is provided.
Typical running time: $\sim 10^{-2}$ sec for 1318 calls of the basic harmonic polylogarithms.
References: [1] G. 't Hooft and M.J.G. Veltman, *Scalar One Loop Integrals*, Nucl. Phys. B **153** (1979) 365.

1 Introduction

Many analytic higher order and multi-leg calculations result in representations based on harmonic polylogarithms (`HPLs`) [2–4]. This function space has been introduced in Ref. [5]. It is formed by the Volterra-iterative integrals over the alphabet

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

with

$$H_{b,\vec{a}}(x) = \int_0^x dy f_b(y) H_{\vec{a}}(y), \qquad H_\emptyset = 1, \tag{2}$$

$$H_{\underbrace{0,\dots,0}_{n}}(x) = \frac{1}{n!} H_0^n(x). \tag{3}$$

By definition all `HPLs` are zero for $x = 0$. The harmonic polylogarithms are generalizations of the classical polylogarithms [6,7] and the Nielsen integrals [8–10]. They obey algebraic [11] and argument relations [5]. Furthermore, they are dual structures, by the Mellin transform

$$\mathbf{M}[f(x)](N) = \int_0^1 dx x^{N-1} f(x), \tag{4}$$

to the nested harmonic sums [12,13]

$$S_{b,\vec{a}}(N) = \sum_{k=1}^{N} \frac{(\mathrm{sign}(b))^k}{k^{|b|}} S_{\vec{a}}(k), \quad S_\emptyset = 1, \qquad b, a_i \in \mathbb{Z} \setminus \{0\}, N \in \mathbb{N} \setminus \{0\}. \tag{5}$$

In the past numerical representations of a series of classical polylogarithms and Nielsen integrals were given in [9] and have been in wide use for decades. The generalization to harmonic polylogarithms has been performed in [14] up to weight $\mathsf{w} = 4$ and in a private version to $\mathsf{w} = 5$. An implementation using the package `Ginac` [15] has been given in [16]. The `Mathematica` packages `HPL` [17] and `HarmonicSums` [18–22] do also allow the numerical evaluation of harmonic polylogarithms. There is also an implementation in `Maple` [23]. In various applications one needs the numerical representation of higher weight harmonic polylogarithms, e.g. to weight $\mathsf{w} = 6$ in Refs. [24–26], which were given in [27]. Numerical implementations of the analytic continuation of harmonic sums to complex arguments have been given in [28,29].

In the foreseeable future fast numerical implementations of harmonic polylogarithms are also needed for higher weights. In the present paper we present a `FORTRAN` implementation for harmonic polylogarithms up to weight $\mathsf{w} = 8$. If one compares the computational times between `FORTRAN` implementations and the ones in `Ginac` [16] or `Mathematica` [17,18] the latter ones are somewhat slower. In many phenomenological and experimental applications users prefer to work with numerical programmes only. The other existing programmes have of course also advantages against pure numeric implementations, since they can be tuned to arbitrary precision and allow analytic functional mappings.

The number of harmonic polylogarithms grows rapidly with the weight. Therefore it is necessary to give numerical representations only after the algebraic relations have been used. Furthermore, it is sufficient to represent the harmonic polylogarithms in the core interval $x \in [0, \sqrt{2} - 1]$ since the representations for the remaining real domain can be given by functional relations.

In Section 2 we describe the argument mappings and the numerical implementation for the harmonic polylogarithms. Technical aspects of the code and its use are discussed in Section 3. In Section 4 we comment on the corresponding numerical implementation for cyclotomic harmonic polylogarithms[1] and Section 5 contains the conclusions. Numerical examples are given in Appendix A. Appendix B contains the list of auxiliary files attached to this paper to perform preparatory calculations in `Mathematica`. Others are subsidiary files used in the initialization of the `FORTRAN`-code.

[1]Beyond these functions various more special functions contribute in analytic calculations in Quantum Field Theories. For recent surveys see [30–32].

2 Mapping Harmonic Polylogarithms and Numerical Implementation

In order to keep the numerical implementation as compact as possible the user is asked to perform mappings on his expressions beforehand into the main interval $x \in [0, \sqrt{2}-1]$. This can be done in `Mathematica` using subsidiary files attched to this paper, which haven been created using the `Mathematica` package `HarmonicSums` [18–22]. For this purpose we work with bases of harmonic polylogarithms for the different weights. These are listed in the files `LIST1.m, ...,` `LIST8.m` attached to this paper. The file `LIST.m` contains the combined basis. The number of basis elements per weight is given in Table 1.

weight	# of basis elements
1	3
2	3
3	8
4	18
5	48
6	116
7	312
8	810

Table 1: Number of basis elements as a function of the weight.

The number of basis elements can be calculated using a formula by Witt, cf. [11], for a 3-letter alphabet

$$N_{\rm bas}(\mathsf{w}) = \frac{1}{\mathsf{w}} \sum_{d|\mathsf{w}} \mu\left(\frac{\mathsf{w}}{d}\right) 3^d, \tag{6}$$

where d are the divisors of w.[2] Here μ denotes the Möbius function for $N \in \mathbb{N}\backslash\{0\}$

$$\mu(N) = \begin{cases} 1 & N = 1 \\ (-1)^k & \text{if } N \text{ is a factor of } k \text{ different primes} \\ 0 & \text{else.} \end{cases} \tag{7}$$

We have e.g.

$$\begin{aligned}
\texttt{LIST1.m} &= \{H[1,x], H[0,x], H[-1,x]\} && (8)\\
\texttt{LIST2.m} &= \{H[0,1,x], H[-1,1,x], H[-1,0,x]\} && (9)\\
\texttt{LIST3.m} &= \{H[0,1,1,x], H[0,0,1,x], H[-1,1,1,x], H[-1,1,0,x], H[-1,0,1,x], H[-1,0,0,x], \\
&\quad H[-1,-1,1,x], H[-1,-1,0,x]\}. && (10)
\end{aligned}$$

[2] We remark that all harmonic polylogarithms up to $\mathsf{w} = 3$ can be expressed in terms of classical polylogarithms, allowing for rational function arguments [6], see also [29], and therefore have a numerical representation using these functions.

The relations of all other `HPLs` are stored in the replacement list

$$\texttt{hrel.m} \qquad [12.8 \text{ Mbyte}] \tag{11}$$

attached to this paper.

Usually one has expressions with main argument x of the harmonic polylogarithms in $x \in]-\infty, +\infty[$. We provide the following algebraic transformation tables of the harmonic polylogarithms, which can be used to map between different kinematic regions. To do this the above basis representation is required:

$$\texttt{LISTOMX.m} \quad x \mapsto -x \qquad [\ 2.6 \text{ Mbyte}] \tag{12}$$

$$\texttt{LISTOOX.m} \quad x \mapsto \frac{1}{x} \qquad [13.1 \text{ Mbyte}] \tag{13}$$

$$\texttt{LISTMTX.m} \quad x \mapsto \frac{1-x}{1+x} \qquad [13.7 \text{ Mbyte}]. \tag{14}$$

These files can be loaded in any `Mathematica` program to carry out the respective transformations. One finally ends up with a representation in the interval

$$x \in [0, \sqrt{2}-1], \tag{15}$$

for which the numeric implementation of `HPOLY` is provided. In course of these mappings also special constants contribute, cf. [33]. They are given by

$$\left\{\ln(2), \zeta_2, \zeta_3, \mathrm{Li}_4\left(\frac{1}{2}\right), \zeta_5, \mathrm{Li}_5\left(\frac{1}{2}\right), \mathrm{Li}_6\left(\frac{1}{2}\right), s_6, \zeta_7, \mathrm{Li}_7\left(\frac{1}{2}\right), s_{7a}, s_{7b}, \mathrm{Li}_8\left(\frac{1}{2}\right), s_{8a}, s_{8b}, s_{8c}, s_{8d}\right\}, \tag{16}$$

with

$$\zeta_k = S_k(\infty), \quad k \geq 2 \tag{17}$$

$$\mathrm{Li}_k(x) = \sum_{l=1}^{\infty} \frac{x^l}{l^k} \tag{18}$$

$$s_6 = S_{-5,-1}(\infty) \tag{19}$$

$$s_{7a} = S_{-5,1,1}(\infty) \tag{20}$$

$$s_{7b} = S_{5,-1,-1}(\infty) \tag{21}$$

$$s_{8a} = S_{5,3}(\infty) \tag{22}$$

$$s_{8b} = S_{-7,-1}(\infty) \tag{23}$$

$$s_{8c} = S_{-5,-1,-1,-1}(\infty) \tag{24}$$

$$s_{8d} = S_{5,-1,1,1}(\infty). \tag{25}$$

Now the Bernoulli-improvement [1], see also [14], of the respective series expansion is performed. To this end we substitute

$$\texttt{term1(x)} \mapsto \texttt{term1(x)}|_{x \to 1-e^{-v}} \tag{26}$$

$$\texttt{term2(x)} \mapsto \texttt{term2(x)}|_{x \to e^{u}-1}, \tag{27}$$

with

$$u = \ln(1+x) \tag{28}$$

$$v \;=\; -\ln(1-x) \tag{29}$$

and expand into series in u and v up to a certain order N_{max}, which we choose $N_{max} = 40$. This representation converges much faster than the corresponding Taylor series. The representations obtained are power series in u and v and logarithms of u and v to a certain power. At weight w there are contributions of up to $O(\ln^{\mathsf{w}-1}(u))$ and of $O(\ln^{\mathsf{w}-2}(v))$. In Appendix A we give one example for illustration.

One stores now the corresponding constants appearing in the expansion for each weight w in a data file. These constants are read into a very compact `FORTRAN` programme. Herewith we obtain a representation of all basis `HPLs` up to $\mathsf{w} = 8$ for $x \in \mathbb{R}$, after arguments from all other regions are mapped using the commands described above. This allows to work with real representations only. Complex valued `HPLs` are already separated into their real and imaginary part by the above transformations.

For the purpose of possible later extensions to even higher weight we list a series of commands of `HarmonicSums`, with which we created the present tables and which can be used also for further extensions. For the arguments $x \in\,]-\infty, 0]$ one first applies the procedure

$$\texttt{TransformH}[H_{\vec{a}}(f(x)), x] \tag{30}$$

of `HarmonicSums`.
Example:

$$\begin{aligned}\texttt{TransformH}[H[-1,0,0,-x],x] &= -H[1,x]H[0,0,-x] + H[0,-x]H[0,1,x] - H[0,0,1,x]\\ &= -\frac{1}{2}H[1,x](H[0,x]+i\pi)^2 + (H[0,x]+i\pi)H[0,1,x]\\ &\quad -H[0,0,1,x].\end{aligned} \tag{31}$$

Here one observes (3) and $H[0,-x] = H[0,x] + i\pi$.
The same command maps $x \in [1,\infty]$ to $x \in [0,1]$.
Example:

$$\texttt{TransformH}\left[H\left[-1,0,0,\frac{1}{x}\right]\right] \;=\; H[-1,0,0,x] - \frac{1}{6}H[0,x]^3. \tag{32}$$

Now only two regions have to be considered:

$$x \in [0, \sqrt{2}-1] \quad \text{and} \quad x \in [\sqrt{2}-1, 1]. \tag{33}$$

The expression in the second region is obtained from the first one by the mapping

$$x \mapsto \frac{1-x}{1+x}. \tag{34}$$

Example:

$$\begin{aligned}\texttt{TransformH}\left[H\left[-1,0,0,\frac{1-x}{1+x}\right]\right] &= \frac{3}{4}\zeta_3 - \frac{1}{6}H[-1,x]^3 - H[-1,x]H[-1,1,x]\\ &\quad +H[-1,-1,1,x] - H[-1,1,1,x].\end{aligned} \tag{35}$$

In the case one is leaving the basis one uses

$$\texttt{ReduceToHBasis}[\textit{expression}] \tag{36}$$

to get back to the basis again. The corresponding substitution rules are tabulated in the file `hrel.m` to weight `w` = 8. The above replacements are illustrated by a `Mathematica` notebook `HTransform.nb` provided in the ancillary files to this paper.
In the case we have `HPLs` over a two-letter alphabet, i.e. $\{0,-1\}$ or $\{0,1\}$, only one branch cut is present, i.e. $[-1,-\infty[$ or $[1,\infty[$. In all other cases one has to consider a cut separated representation of the respective `HPL`. This cut separation is performed by using the command

$$\texttt{HSeparateCuts}[H_{\vec{a}}(x)]. \tag{37}$$

<u>Example:</u>

$$\begin{aligned} \texttt{HSeparateCuts}[H[-1,1,1,x]] &= \{\texttt{term1},\texttt{term2}\} \\ &= \left\{-\frac{1}{2}\ln^2(2)H[-1,x]+H[-1,1,1,x],\frac{1}{2}\ln^2(2)H[-1,x]\right\}. \end{aligned} \tag{38}$$

This command is needed to obtain the representations (26,27). The sum of the two terms yields the original expression again.

3 Description of the code

The code `HPOLY` provides the calculation of all `HPLs`, using the basis representation and argument mappings described in Section 2 in the region $x \in [0, \sqrt{2}-1]$. It shall be compiled using

```
gfortran hpoly.f.
```

in `LINUX`. There is a user-routine `UHPOLY` through which the corresponding calculations can be made. Through the initialization routine `UHPOLYIN` the maximal weight of the calculation is chosen setting the variable `IW`. If the calculation of an `HPL` is intended at a higher weight than `w` = 8, the code cannot perform the calculation. Choosing `IW = 8` the initialization time is longest with ~ 0.27 sec for reading in the constant tables. Through `UHPOLY` the user can arrange special output formats. Only the routines `UHPOLYIN` and `UHPOLY` are free to be modified by the user.

A built in logic checks whether the required `HPL` is contained in the basis and whether the variable is in the correct range. The routine `HPOLYIN` performs necessary initializations.

The main routine is

```
      PROGRAM HPOLY
*
*---------------------------------------------------------------
*
*     numerical evaluation of HPLs up to weight w = 8
*
*     J. Bluemlein, 17.9.2018
*
*---------------------------------------------------------------
*
      WRITE(6,*)
     &'**** HPOLY: calculation of HPLS up to w=8 ****'
      CALL UHPOLYIN
```

```
*
      CALL HPOLYIN
      WRITE(6,*) '**** HPOLY: initialization finished ****'
*
      CALL UHPOLY
*
      WRITE(6,*) '**** HPOLY: finished ****'
      STOP
      END
      SUBROUTINE UHPOLYIN
*
*---------------------------------------------------------------
*
      IMPLICIT NONE
      INTGER   IW
      COMMON/WEIGHT/ IW
*
      IW = 8
*
      RETURN
      END
      SUBROUTINE UHPOLY
*
*---------------------------------------------------------------
*
      IMPLICIT NONE
*
      REAL*8   H1,H2,H3,H4,H5,H6,H7,H8
      REAL*8   T(8),X
      INTEGER  K
*
      EXTERNAL H1,H2,H3,H4,H5,H6,H7,H8
*
      CALL CPU_TIME(START)
      X   =  0.3D0
      CALL   HPOLYC(X)
*
      T(1)  =  H1(-1,X)
      T(2)  =  H2(0,1,X)
      T(3)  =  H3(-1,0,0,X)
      T(4)  =  H4(-1,-1,1,0,X)
      T(5)  =  H5(-1,-1,1,0,1,X)
      T(6)  =  H6(-1,0,-1,1,1,1,X)
      T(7)  =  H7(-1,-1,1,1,0,1,0,X)
      T(8)  =  H8(-1,0,-1,0,-1,0,1,1,X)
*
      DO 1 K = 1,8
      WRITE(6,*) 'K, X , T=', K,X,T(K)
```

```
1     CONTINUE
      CALL CPU_TIME(END)
      WRITE(6,*) 'CPU TIME=', END-START, '   SEC'
*
      RETURN
      END
*---------------------------------------------------------------
```

Running the above example produces the following output values:

HPL	value	accuracy
`H[-1,x]`	0.26236426446749106e-0	7.96e-18
`H[0,1,x]`	0.32612951007547608e-0	1.05e-17
`H[-1,0,0,x]`	0.81699704232693138e-0	1.20e-16
`H[-1,-1,1,0,x]`	−0.10536957058865759e-0	1.20e-16
`H[-1,-1,1,0,1,x]`	0.27247014022231675e-3	3.97e-17
`H[-1,0,-1,1,1,1,x]`	0.45411840144185533e-5	8.46e-19
`H[-1,-1,1,1,0,1,0,x]`	−0.80698691040978487e-4	3.86e-18
`H[-1,0,-1,0,-1,0,1,1,x]`	0.57153046109648109e-6	3.47e-18

Table 2: The result of a test calculation of the code `HPOLY.f` with $x = 0.3$.

The following output is obtained by running the code default:

```
**********************************************
*  HPOLY.f: J. Bluemlein 17.09.2018 **********
**********************************************
**** HPOLY: calculation of HPLS up to w=8 ****
**** HPLs up to weight w =           8
READ-IN TIME=  0.26383499999999999       SEC
**** HPOLY: initialization finished ****
K,X,T=           1  0.29999999999999999       0.26236426446749106
K,X,T=           2  0.29999999999999999       0.32612951007547608
K,X,T=           3  0.29999999999999999       0.81699704232693138
K,X,T=           4  0.29999999999999999       -1.0536957058865759E-002
K,X,T=           5  0.29999999999999999        2.7247014022231675E-004
K,X,T=           6  0.29999999999999999        4.5411840144185533E-006
K,X,T=           7  0.29999999999999999       -8.0698691040978487E-005
K,X,T=           8  0.29999999999999999        5.7153046109648109E-007
CPU TIME=   2.2499999999997522E-004    SEC
**********************************************
**** HPOLY: calculation    finished ****
**********************************************
```

The absolute accuracy has been determined comparing with the corresponding result obtained using `Ginac` [16]. We have compared the numerical values produced by `HPOLY` and using `Ginac` forming the difference in a `Maple` programme, which can handle floating point number inputs straightforwardly.

4 Cyclotomic Harmonic Polylogarithms

In the following we would like to make a few brief remarks on the numerical evaluation of cyclotomic harmonic polylogarithms [20].

These functions emerge in massive calculations from 3-loops onward. Usually their main argument x is located in the interval $[-1,1]$. This is going to be the range we are considering in the following. The new letters are of cyclotomy 3,4 and 6 :

$$\left\{\frac{1}{1+x+x^2}, \quad \frac{x}{1+x+x^2}, \quad \frac{1}{1+x^2}, \quad \frac{x}{1+x^2}, \quad \frac{1}{1-x+x^2}, \quad \frac{x}{1-x+x^2}\right\}. \tag{39}$$

Here cyclotomy labels the degree of the highest cyclotomic polynomial, cf. [20]. While here the transformations $x \mapsto 1/x$ and $x \mapsto -x$ are class preserving, the transformation $x \mapsto (1-x)/(1+x)$ is not. There is also no reason, why the above Bernoulli-improvement should accelerate the convergence of the expansions, given the structure of the above letters.

Yet one would like to have a series representation in the interval $[-1,1]$, see e.g. Ref. [25,26]. One possibility is given by Taylor series in x around $x=0,1$ and $x=-1$, which can be calculated analytically using the command

$$\texttt{HarmonicSumsSeries}[\texttt{f}(\texttt{x}), \{\texttt{x}, 0, \texttt{n}\}] \tag{40}$$

of `HarmonicSums`, where `n` denotes the highest power in the expansion. The series are intended to hold at a convergence radius of $\rho = 1/2$ at double precision, i.e. $53 \cdot \ln_{10}(2) \approx 15.955$ digits [34].

x	`HPL`	run time `Fortran` [sec]	run time `Ginac` [sec]	accuracy
-1.0	$-0.19456824628601374558\text{e}{+0}$	6.0e-6	2.4e-4	5.5e-18
-0.8	$-0.10723997730030421271\text{e}{+0}$	6.0e-6	2.9e-4	4.5e-18
-0.4	$-0.16724871547938951751\text{e}{-1}$	8.0e-6	3.3e-4	5.2e-18
0.4	$0.25064429005154578702\text{e}{-1}$	8.0e-6	2.7e-4	2.3e-18
0.8	$0.19221120777067257053\text{e}{+0}$	8.0e-6	6.2e-3	6.0e-18
1.0	$0.34039020776540250510\text{e}{+0}$	2.9e-5	6.0e-3	1.5e-17

Table 3: Example of the numerical calculation of an cyclotomic `HPL` `H[{6,1},0,-1,x]` comparing the run times in `FORTRAN` and `Ginac`.

The expansion around $x=-1$ may be performed by mapping

$$\left\{\frac{1}{1+x+x^2} \leftrightarrow \frac{1}{1-x+x^2}, \quad \frac{x}{1+x+x^2} \leftrightarrow -\frac{x}{1-x+x^2}\right\} \tag{41}$$

more effectively. In Table 3 we illustrate the run times needed to calculate the cyclotomic harmomic polylogarithm

$$\mathtt{H}[\{6,1\},0,-1,\mathtt{x}] = \int_0^x dy_1 \frac{y_1}{1-y_1+y_1^2} \int_0^{y_1} \frac{dy_2}{y_2} \int_0^{y_2} \frac{dy_2}{1+y_2}. \tag{42}$$

We note that some of the cyclotomic HPLs can be complex for $x \in [-1,0[$.
The accuracy of the respective representation can be tested comparing the numerical result with the complex-valued representations provided by Ginac, cf. Ref. [16]. To have these explicit series representations is of importance since they usually perform faster than the algorithm based on a Hölder convolution in [16]. However, the latter one has the advantage that it can be extended to arbitrary precision.

In Table 3 we summarize the performance for a series of values using both implementations by considering the example of H({6,1},0,-1,x). The function is graphically illustrated in Figure 1.

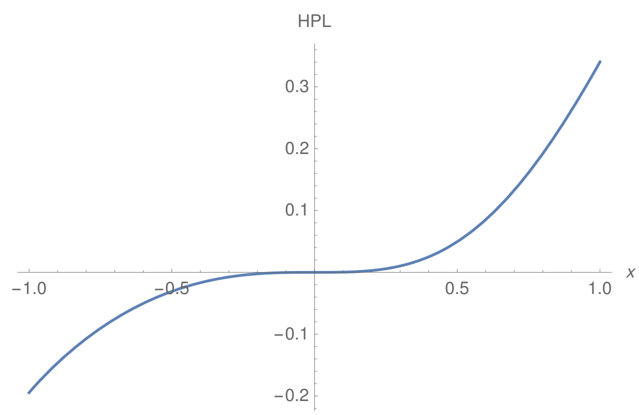

Figure 1: The function H[{6,1},0,-1,x] in the region $x \in [-1,1]$.

In the case of the planar contributions to the 3-loop massive Form factors [25,26] 206 real-valued cyclotomic polylogarithms up to w=6 contribute. Their numerical representation is given in Ref. [26].[3]

5 Conclusions

We presented a compact FORTRAN code which calculates the HPLs numerically up to weight $\mathsf{w} = 8$. Computer-readable replacement files are provided to allow the analytic representation of a corresponding polynomial of HPLs over the chosen algebraic basis, mapping the emerging arguments $x \in \mathbb{R}$ into the interval $x \in [0, \sqrt{2}-1]$ in a Mathematica session during a preparatory step. For the implementation of the HPLs we used the Bernoulli-improvement [1]. The numerical accuracy of the representation is $\sim 10^{-15}$ or better. We also compared run times with corresponding implementations in Ginac. Because we limit the accuracy to about double precision, the FORTRAN implementation is faster. Numerical representations using Mathematica are usually slower, cf. Table 4. This is of advantage for larger scale phenomenological and experimental applications. The present code extends earlier work of the authors of Ref. [14] to far higher weight.

[3]In Refs. [35] the numerical representations are based on Ginac.

We briefly commented on the numerical implementation of cyclotomic harmonic polylogarithms, which are important in massive higher loop calculations, as e.g. for the massive Form factors from 3-loop order [25, 26, 35]. Here, precise Taylor series around $x = 0, 1, -1$ providing sufficient convergence in a circle of radius $\rho = 1/2$ within $x \in [-1, 1]$ serve the purpose. Detailed representations are given in Ref. [26].

The use of the `FORTRAN`-code `HPOLY.f` and of the attached subsidiary files requires the citation of the present paper. Discovered bugs shall be reported to `Johannes.Bluemlein@desy.de`.

A Numerical examples

A.1 Harmonic Polylogarithms

In the following we show the explicit representation for one example, `H[-1,1,0,x]` and give some numerical illustration. This harmonic polylogarithm is represented by

$$
\begin{aligned}
&\mathtt{H}[-1,1,0,\mathtt{x}] \\
&\quad = 1.515614213984058527653439704 78u - 0.17328679513998632735430803 0365u^2 \\
&\quad -9.6270441744436848530171127 9803e\text{-}3u^3 + 4.8135220872218424265085563 9902e\text{-}5u^5 \\
&\quad -5.4575080353989143157693383 2088e\text{-}7u^7 + 7.9588658849567500438302850 5128e\text{-}9u^9 \\
&\quad -1.3155150223069008336909562 0682e\text{-}10u^{11} + 2.3478983949832706952392594 4587e\text{-}12u^{13} \\
&\quad -4.4171749833259796003054085 0887e\text{-}14u^{15} + 8.6380416926308759809173132 4426e\text{-}16u^{17} \\
&\quad -1.7401768147431913540605806 8735e\text{-}17u^{19} + 3.5892954916514242379052191 0252e\text{-}19u^{21} \\
&\quad -7.5465410635862794583439948 632e\text{-}21u^{23} + 1.6120829755323789347574740 235e\text{-}22u^{25} \\
&\quad -3.4901313655620459400434229 4876e\text{-}24u^{27} + 7.6429868973465658122285358 3268e\text{-}26u^{29} \\
&\quad -1.6903492989052459770495696 2836e\text{-}27u^{31} + 3.7708195596930392935062854 4829e\text{-}29u^{33} \\
&\quad -8.4760394848813392850687682 4739e\text{-}31u^{35} + 1.9181208713816686572020940 8063e\text{-}32u^{37} \\
&\quad -4.3668900310005976780897072 5046e\text{-}34u^{39} \\
&\quad -1.5156142139840585276534397 0478v + 2.4575382856409954559051561 3688e\text{-}1v^2 \\
&\quad -7.2844336941139778276349149 6943e\text{-}3v^3 - 4.6652224716482883657069110 26793e\text{-}3v^4 \\
&\quad +7.1818006008597602017657169 7142e\text{-}4v^5 + 1.0998660373605051065035495 605e\text{-}5v^6 \\
&\quad -1.1545613692664445964851812 9105e\text{-}5v^7 + 2.2668351288044216255466542 194e\text{-}7v^8 \\
&\quad +2.1015321013714847072831050 1335e\text{-}7v^9 - 1.1468791479418896605244710 8046e\text{-}8v^{10} \\
&\quad -3.8990842757621557606324231 5571e\text{-}9v^{11} + 3.6613810706393983341317228 4743e\text{-}10v^{12} \\
&\quad +7.0646198089444308450978101 7833e\text{-}11v^{13} - 1.0170196199844264460947848 7375e\text{-}11v^{14} \\
&\quad -1.2038274227985936691980952 6002e\text{-}12v^{15} + 2.6261544211218086487573803 3e\text{-}13v^{16} \\
&\quad +1.8069183954041747178090216 0508e\text{-}14v^{17} - 6.4503615576441778340886025 6621e\text{-}15v^{18} \\
&\quad -1.9433133319674504101393838 6874e\text{-}16v^{19} + 1.5206334425075410405765398 0337e\text{-}16v^{20} \\
&\quad -5.8251089793978778703671657 2212e\text{-}19v^{21} - 3.4490796854322034166778519 2179e\text{-}18v^{22} \\
&\quad +1.2682944097797912296020789 4152e\text{-}19v^{23} + 7.5115131718216306958455248 7431e\text{-}20v^{24} \\
&\quad -5.4014857348969579203063854 0445e\text{-}21v^{25} - 1.5599851764915693418924142 3249e\text{-}21v^{26} \\
&\quad +1.7684948178785712544928690 8312e\text{-}22v^{27} + 3.0452339831427789586477860 4675e\text{-}23v^{28}
\end{aligned}
$$

$$-5.1317280923991163476597539210 6e\text{-}24v^{29} - 5.4195445218193828742680705313 2e\text{-}25v^{30}$$

$$+1.3815554586682880684047703292 3e\text{-}25v^{31} + 8.1368615251635615959596762332 8e\text{-}27v^{32}$$

$$-3.5180918888246728935421940040 5e\text{-}27v^{33} - 7.4706318697360743524018936265 5e\text{-}29v^{34}$$

$$+8.5461501918853396299109210199 7e\text{-}29v^{35} - 1.0816037809007587442923675520 2e\text{-}30v^{36}$$

$$-1.9853612931962075648278341775 7e\text{-}30v^{37} + 9.1894371017749382860459822819 2e\text{-}32v^{38}$$

$$+4.4008695848170510027579730992 9e\text{-}32v^{39} - 3.6546241595922893462374963063 5e\text{-}33v^{40}$$

$$+\Big[0.69314718055994530941723212145 8u + 0.50000000000000000000000000000 0u^2$$

$$+0.33333333333333333333333333333 3u^3 + 0.25000000000000000000000000000 0u^4$$

$$+0.21666666666666666666666666666 7u^5 + 0.20833333333333333333333333333 3u^6$$

$$+0.21468253968253968253968253968 3u^7 + 0.23229166666666666666666666666 7u^8$$

$$+0.26065365961199294532627865961 2u^9 + 0.30083498677248677248677248677 2u^{10}$$

$$+0.35510166145582812249478916145 6u^{11} + 0.42691950507054673721340388007 1u^{12}$$

$$+0.52115855241723297278852834408 4u^{13} + 0.64446245064734648067981401314 7u^{14}$$

$$+0.80579441070913458479596045733 6u^{15} + 1.0172011502678432861964607996 4u^{16}$$

$$+1.2948626956570395881884366167 0u^{17} + 1.6605262130280236300132684612 4u^{18}$$

$$+2.1434610451383272815264024678 3u^{19} + 2.7831245581437653711079863948 4u^{20}$$

$$+3.6327999983807456302536222770 5u^{21} + 4.7645659474593963795711111555 9u^{22}$$

$$+6.2760925631369577080454104148 1u^{23} + 8.2999469822479763262492088621 7u^{24}$$

$$+11.016348962856034834470879059 6u^{25} + 14.670675708382778585026243911 7u^{26}$$

$$+19.597510269420384214844444545 2u^{27} + 26.253714388073084237487898116 1u^{28}$$

$$+35.263958479995971034498136006 1u^{29} + 47.483462153124195630908241362 3u^{30}$$

$$+64.084532445289699278647120404 5u^{31} + 86.676034834225617031901395861 2u^{32}$$

$$+117.46847436960444360793564330 8u^{33} + 159.50229217300765438880376831 8u^{34}$$

$$+216.96384216108234222913389716 3u^{35} + 295.62306696367544297991673479 1u^{36}$$

$$+403.44020659002210467757318730 1u^{37} + 551.40743874566253280258764624 9u^{38}$$

$$+754.71725034213156788784855737 8u^{39} + 1034.3854926248501577528431023 2u^{40}\Big]$$

$$\times H[0,u]$$

$$+\Big[6.9314718055994530941723212145 8e\text{-}1v - 1.9314718055994530941723212145 8e\text{-}1v^2$$

$$+2.6480513893278642750565454791 5e\text{-}2v^3 - 9.0944560660741853533479824635 8e\text{-}4v^4$$

$$-2.3269096673502989512651484394 6e\text{-}4v^5 + 2.0486991859965570215061919795 8e\text{-}5v^6$$

$$+3.2724453269380955944542812066 9e\text{-}6v^7 - 4.7669749471816811303175206073 6e\text{-}7v^8$$

$$-4.7944567085726659529232121874 8e\text{-}8v^9 + 1.0999526811306392227536487035 9e\text{-}8v^{10}$$

$$+6.3007973089991856182953345505 8e\text{-}10v^{11} - 2.4918346059372030398198902277 e\text{-}10v^{12}$$

$$-5.3400788391886541481868737171 9e\text{-}12v^{13} + 5.5156739926803903227757686507 e\text{-}12v^{14}$$

$$-6.2885778619766279632084193422 1e\text{-}14v^{15} - 1.1872510579164206583971827922 4e\text{-}13v^{16}$$

$$+5.2657813263214822480874274162 3e\text{-}15v^{17} + 2.4684592925708782371421863907 4e\text{-}15v^{18}$$

$$-1.9904199690311149991339847040 5e\text{-}16v^{19} - 4.9037064983133000546399351957 8e\text{-}17v^{20}$$

$$+6.1270186457563466938729356423 9e\text{-}18v^{21} + 9.1272104720065852398251456019 8e\text{-}19v^{22}$$

$$-1.702881803015154903381436824 83e\text{-}19v^{23} - 1.527638682506556982830765780 6e\text{-}20v^{24}$$
$$+4.429444490901249294921598791 93e\text{-}21v^{25} + 2.051343503996898530051534539 42e\text{-}22v^{26}$$
$$-1.094921245042866236362328611 58e\text{-}22v^{27} - 1.103999577246476421081992278 76e\text{-}24v^{28}$$
$$+2.588127128844959001315567989 53e\text{-}24v^{29} - 6.101047484717702670830482142 69e\text{-}26v^{30}$$
$$-5.854272176676917791893842556 22e\text{-}26v^{31} + 3.398959819454887474508903589 33e\text{-}27v^{32}$$
$$+1.261935210675019653990160099 55e\text{-}27v^{33} - 1.225505440383377799762902045 37e\text{-}28v^{34}$$
$$-2.564605844301315633948582211 59e\text{-}29v^{35} + 3.761557357066875630807273512 39e\text{-}30v^{36}$$
$$+4.801460800593487936291862052 66e\text{-}31v^{37} - 1.054005775546030831384575245 55e\text{-}31v^{38}$$
$$\left.-7.836896861477280077275554622 51e\text{-}33v^{39} + 2.770928876212783774307216641 71e\text{-}33v^{40}\right]$$
$$\times H[0,v] \tag{43}$$

for $x \in [0, \sqrt{2}-1]$.

x	HPL	run time Fortran [sec]	run time Ginac [sec]	run time Mathematica [sec]
-20.0	5.369919763979762	2.5e-5	0.02923	–
	-18.46370249603318 i			
-0.9	-1.652038279906588	2.0e-5	0.01917	3.45
	$+3.344002738868969$ i			
-0.2	-0.067890106575246	8.6e-5	0.00133	0.94
	$+0.068215824899983$ i			
0.2	-0.058464914759637	5.4e-5	0.00189	0.84
0.9	-0.550223509450311	8.2e-5	0.01310	1.05
50.0	-18.95831087429180	3.5e-5	0.01260	2.08

Table 4: Some numerical values for `H[-1,1,0,x]` and the run times in the `FORTRAN` implementation, by `Ginac`, and `Mathematica`.

The following mappings hold :

$$\begin{aligned}
H[-1,1,0,z] &= \frac{1}{2}\ln(2)\zeta_2 - \zeta_3 + \zeta_2 H[-1,x] + \frac{1}{6}H[-1,x]^3 - H[-1,x]H[0,-1,x] \\
&\quad -H[-1,x]H[0,1,x] + H[-1,-1,1,x] + 2H[0,-1,-1,x] + H[0,-1,1,x] \\
&\quad +H[0,1,-1,x] \quad \text{for} \quad z \in [\sqrt{2}-1,1], \quad z = \frac{1-x}{1+x}. \qquad (44)\\
H[-1,1,0,-x] &= H[-1,x]H[0,-x]H[1,x] - H[0,-x]H[-1,1,x] - H[1,x]H[0,-1,x] \\
&\quad +H[0,-1,1,x] \qquad (45)\\
H\left[-1,1,0,\frac{1}{x}\right] &= 3\ln(2)\zeta_2 + \frac{3}{4}\zeta_3 - 2\zeta_2 H[-1,x] + 2\zeta_2 H[0,x] - \frac{1}{2}H[-1,x]H[0,x]^2 + \frac{1}{6}H[0,x]^3
\end{aligned}$$

$$-H[0,x]H[-1,1,x] + H[0,x]H[0,-1,x] + H[-1,x]H[0,1,x]$$
$$+H[0,x]H[0,1,x] - H[0,0,-1,x] - 2H[0,0,1,x] - H[0,1,-1,x]. \qquad (46)$$

Here $H[0,-x] = H[0,x] + i\pi$ and one can see that the function is real for $x \geq 0$ and complex for $x \leq 0$.

In Table 4 we present some numerical results and compare the run times of `HPOLY`, `Ginac` [16], and `Mathematica`. In the latter case the harmonic polylogarithm is represented by a double-integral because of its last letter being 0 and has been computed using the `Mathematica`-command `NIntegrate`. In Figure 2 the behaviour of $H[-1,1,0,x]$ in the range $x \in [-3,3]$ is illustrated for the real and imaginary part of this function.

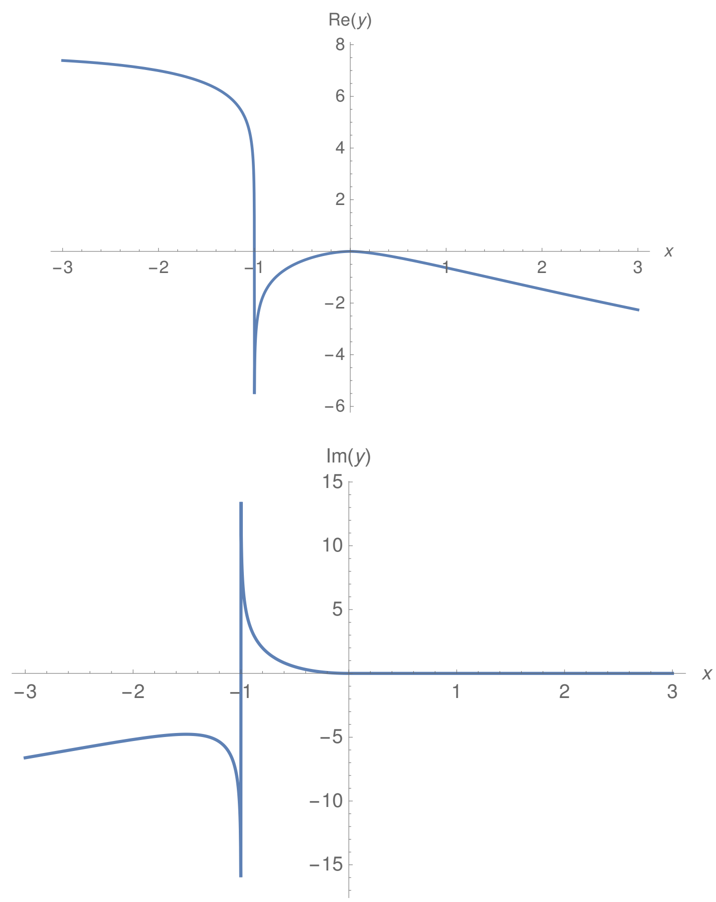

Figure 2: Real and imaginary part of the function $H[-1,1,0,x]$.

We have also measured the run times for all `HPLs` in `HPOLY` and `Ginac` at $x = 0.3$, yielding $9 \cdot 10^{-3}$ sec and 2.7 sec, respectively.

A.2 Cyclotomic Harmonic Polylogarithms

We discuss the representation of the cyclotomic harmonic polylogarithm $H[\{6,1\},0,-1,x]$, i.e. a real representation w.r.t. the indices of this iterated integral, and give some numerical illustrations. The treatment is very similar to the case of the usual harmonic polylogarithms, since only more letters (regular on $x \in [-1,1]$) are added to the alphabet. In the following we display the first 40 expansion coefficients, which show that the series converges only slowly, requiring the expansion coefficients $|\xi| \leq 1/2$ to reach double precision, with $\xi \in x, x+, x-$. One obtains

$$
\begin{aligned}
&\texttt{H}[\{\texttt{6},\texttt{1}\},\texttt{0},-\texttt{1},\texttt{x}] \\
&\quad 0.33333333333333333333333333333333x^3 + 0.18750000000000000000000000000000x^4 \\
&\quad -0.027777777777777777777777777777778x^5 - 0.15856481481481481481481481481481 5x^6 \\
&\quad -0.11035714285714285714285714285 7x^7 + 0.018888888888888888888888888888889x^8 \\
&\quad +0.10489103048626858150667674477 2x^9 + 0.077728316326530612244897959183 7x^{10} \\
&\quad -0.014035493827160493827160493827 2x^{11} - 0.078472799613672629545645418661 3x^{12} \\
&\quad -0.059924515419645289775159905030 0x^{13} + 0.011122175033159881644730129578 6x^{14} \\
&\quad +0.062709754045529381304717080052 9x^{15} + 0.048739613712648490370768093045 8x^{16} \\
&\quad -0.0091981204021247311290601333891 4x^{17} - 0.052228228679916459729239542019 4x^{18} \\
&\quad -0.041067361146837847290918168319 8x^{19} + 0.0078370917347745378422890069259 8x^{20} \\
&\quad +0.044751989504185853817403290662 3x^{21} + 0.035479542949655098787258589335 8x^{22} \\
&\quad -0.0068249678695367666143142108929 5x^{23} - 0.039149596733093214345495459272 4x^{24} \\
&\quad -0.031229028057065981847565168014 2x^{25} + 0.0060434041944798648842942581181 3x^{26} \\
&\quad +0.034794600388263927154843700423 2x^{27} + 0.027887371838779326565080903393 9x^{28} \\
&\quad -0.0054219329960283460736179535193 8x^{29} - 0.031311932619157493753674905719 4x^{30} \\
&\quad -0.025191382614445484830491718879 8x^{31} + 0.0049160627004938647423091492248 7x^{32} \\
&\quad +0.028463256066361831650763716902 0x^{33} + 0.022970496508062846256509114193 2x^{34} \\
&\quad -0.0044963512934359228267083623819 6x^{35} - 0.026089839679097451625203156627 8x^{36} \\
&\quad -0.021109340720179424720934101934 1x^{37} + 0.0041425530753315247077494805816 8x^{38} \\
&\quad +0.024081873941004007767901304416 2x^{39} + 0.019527088651523377383890241707 5x^{40}
\end{aligned}
\tag{47}
$$

for $|x| \leq 1/2$,

$$
\begin{aligned}
&\texttt{H}[\{\texttt{6},\texttt{1}\},\texttt{0},-\texttt{1},\texttt{x}] = \\
&\quad 0.34039020776540250509742089786 1 - 0.82246703342411321823620758332 3x_- \\
&\quad +0.34657359027997265470861606072 9x_-^2 + 0.30634687456802862431494121468 4x_-^3 \\
&\quad +0.038008894929370752989513208921 1x_-^4 - 0.15662012847331966068764136369 8x_-^5 \\
&\quad -0.15659754129645754125563132115 2x_-^6 - 0.022555446077417556979133327178 6x_-^7 \\
&\quad +0.097651930791825178576795622872 2x_-^8 + 0.10432536906718006024234286609 6x_-^9 \\
&\quad +0.015764632630060188374015872554 8x_-^{10} - 0.071028009794858302835551420662 2x_-^{11} \\
&\quad -0.078247079006063564622013833438 9x_-^{12} - 0.012127781776882040968923561010 2x_-^{13} \\
&\quad +0.055807284056778666521682595094 6x_-^{14} + 0.062597491501945028702921466740 7x_-^{15} \\
&\quad +0.0098537541276526269140623389120 7x_-^{16} - 0.045958967626193286159946415100 9x_-^{17} \\
&\quad -0.052164587690322775378333673163 1x_-^{18} - 0.0082979029736033445549192916858 5x_-^{19} \\
&\quad +0.039065120479670786326990872756 3x_-^{20} + 0.044712502884208783034795051317 3x_-^{21} \\
&\quad +0.0071663703857813758247253168664 8x_-^{22} - 0.033969670139455842621620841414 3x_-^{23} \\
&\quad -0.039123440091957043498268010620 4x_-^{24} - 0.0063064059693398551290504478047 5x_-^{25} \\
&\quad +0.030050092802544666048969095220 2x_-^{26} + 0.034776391187047096424649534140 4x_-^{27}
\end{aligned}
$$

$$+0.0056307196129038316633257351954 0x_-^{28} - 0.026941462513776729936236874532 8x_-^{29}$$
$$-0.031298752068857911291432361843 3x_-^{30} - 0.005085811263499876270197327204 81x_-^{31}$$
$$+0.024415700403005469398077041166 5x_-^{32} + 0.028453410971641587186351198961 4x_-^{33}$$
$$+0.004637063210816573163000689641 77x_-^{34} - 0.022322926082757678848638108262 3x_-^{35}$$
$$-0.026082293390675943348953686115 0x_-^{36} - 0.004261085112644313086904886927 22x_-^{37}$$
$$+0.020560589813065338424990930237 7x_-^{38} + 0.024075963129854281967031354673 6x_-^{39}$$
$$+0.003941503729195788893687838083 96x_-^{40} \tag{48}$$

for $x \in [1/2, 1]$ and

$\texttt{H[\{6,1\},0,-1,x]} =$

$$-0.19456824628601374557862520308 2$$
$$+\Big(-0.54831135561607547882413838888 2x_+ + 0.25000000000000000000000000000 0x_+^2$$
$$+0.10721978025363801616564500617 2x_+^3 + 0.27174094449487771383493013888 3e\text{-}1x_+^4$$
$$-0.25750508615077564967049604941 3e\text{-}2x_+^5 - 0.94345633733648573880303590536 3e\text{-}2x_+^6$$
$$-0.81690505245811073178868969094 2e\text{-}2x_+^7 - 0.52079527798773830519862265894 0e\text{-}2x_+^8$$
$$-0.27706031398482974103669022252 6e\text{-}2x_+^9 - 0.12685193802712390083184822079 4e\text{-}2x_+^{10}$$
$$-0.50366772677932306378040084984 7e\text{-}3x_+^{11} - 0.18023549828298563192318553017 8e\text{-}3x_+^{12}$$
$$-0.73366875698581618767943809822 7e\text{-}4x_+^{13} - 0.51734276725738820375518186819 2e\text{-}4x_+^{14}$$
$$-0.52976854749494559536326693896 8e\text{-}4x_+^{15} - 0.54152382304817587224611835109 1e\text{-}4x_+^{16}$$
$$-0.50495662348735057943225713681 8e\text{-}4x_+^{17} - 0.43505742279888628419264262517 7e\text{-}4x_+^{18}$$
$$-0.35597691275089331670248618527 6e\text{-}4x_+^{19} - 0.28374769650669256623293406646 8e\text{-}4x_+^{20}$$
$$-0.22486263708697558023155430400 3e\text{-}4x_+^{21} - 0.17965424413049071925656806372 4e\text{-}4x_+^{22}$$
$$-0.14574327941654360560901473290 2e\text{-}4x_+^{23} - 0.12021524679258425632986718152 8e\text{-}4x_+^{24}$$
$$-0.10060351673805668079134028973 3e\text{-}5x_+^{25} - 0.85134100682393265466152193830 3e\text{-}5x_+^{26}$$
$$-0.72634982732279915140847248505 8e\text{-}5x_+^{27} - 0.62354950067975963294139463630 0e\text{-}5x_+^{28}$$
$$-0.53802864323223201465060682913 5e\text{-}5x_+^{29} - 0.46638127803508785353877428235 9e\text{-}5x_+^{30}$$
$$-0.40607045771597820029219274228 5e\text{-}5x_+^{31} - 0.35509771118623153990722217026 5e\text{-}5x_+^{32}$$
$$-0.31184096698703070217910712244 2e\text{-}5x_+^{33} - 0.27497122578181557160031721945 9e\text{-}5x_+^{34}$$
$$-0.24340065829699314139140609340 5e\text{-}5x_+^{35} - 0.21624202655894861049808364725 2e\text{-}5x_+^{36}$$
$$-0.19277319045081778835883875768 3e\text{-}5x_+^{37} - 0.17240588375481390221963447833 7e\text{-}5x_+^{38}$$
$$-0.15465918552502254876910901407 4e\text{-}5x_+^{39} - 0.13913782954652299375406793006 6e\text{-}5x_+^{40}$$
$$+\Big[-0.16666666666666666666666666666 7x_+^2 - 0.55555555555555555555555555555 6e\text{-}1x_+^3$$
$$+0.16666666666666666666666666666 7e\text{-}1x_+^5 + 0.16666666666666666666666666666 7e\text{-}1x_+^6$$
$$+0.11904761904761904761904761904 8e\text{-}1x_+^7 + 0.72420634920634920634920634920 6e\text{-}2x_+^8$$
$$+0.40123456790123456790123456790 1e\text{-}2x_+^9 + 0.21428571428571428571428571428 6e\text{-}2x_+^{10}$$
$$+0.11904761904761904761904761904 8e\text{-}2x_+^{11} + 0.74855699855699855699855699855 7e\text{-}3x_+^{12}$$
$$+0.54945054945054945054945054945 1e\text{-}3x_+^{13} + 0.44895580609866324152038437752 7e\text{-}3x_+^{14}$$
$$+0.38239538239538239538239538239 5e\text{-}3x_+^{15} + 0.32675657675657675657675657675 7e\text{-}3x_+^{16}$$
$$+0.27676571794218853042382454147 2e\text{-}3x_+^{17} + 0.23265586991077187155618528167 5e\text{-}3x_+^{18}$$
$$+0.19519953730480046269519953730 5e\text{-}3x_+^{19} + 0.16437574293611445314231691940 7e\text{-}3x_+^{20}$$

$$
\begin{aligned}
&+0.13944971928165205476129845877 7e\text{-}3x_+^{21} + 0.11937530056021471726509722569 4e\text{-}3x_+^{22}\\
&+0.13113447776255691183676107623 0e\text{-}3x_+^{23} + 0.89789664850238278694329306794 0e\text{-}4x_+^{24}\\
&+0.78731210310157678578731210310 2e\text{-}4x_+^{25} + 0.69443019019522741941123129190 7e\text{-}4x_+^{26}\\
&+0.61564651410251486030589942686 2e\text{-}4x_+^{27} + 0.54829996359811947374203397894 3e\text{-}4x_+^{28}\\
&+0.49037073895387772504941474636 2e\text{-}4x_+^{29} + 0.44027918049111786608363397368 3e\text{-}4x_+^{30}\\
&+0.39675943750646636894514653428 1e\text{-}4x_+^{31} + 0.35878062963969632377164354709 0e\text{-}4x_+^{32}\\
&+0.32549538428283547702447878028 5e\text{-}4x_+^{33} + 0.29620352809046470158529904618 7e\text{-}4x_+^{34}\\
&+0.27032445862260314010626952485 3e\text{-}4x_+^{35} + 0.24737508837810504614174585840 1e\text{-}4x_+^{36}\\
&+0.22695190477074028457230294841 4e\text{-}4x_+^{37} + 0.20871636627585874942044108665 8e\text{-}4x_+^{38}\\
&+0.19238308386631343514416624206 1e\text{-}4x_+^{39} + 0.17771031017051445367198606128 9e\text{-}4x_+^{40}\Big]\\
&\times H[-1,x]\Big) \qquad (49)
\end{aligned}
$$

for $x \in [-1, -1/2]$. Here we used the notation $x_\pm = (1 \pm x)$.

B List of the attached files

The following files are attached to this paper. The analytic files are given in `Mathematica` format and can be easily converted into `Maple` or `FORM` [36] format.

File name	contents
`LISTn.m`	Bases per weight `n`, `n = 1...8`
`LIST.m`	Total basis
`hrel.m`	Replacement rules to basis for all `HPLs` up to weight `w`=8.
`LISTOMX.m`	Replacement rules: $x \mapsto -x$ for the basis elements
`LISTOOX.m`	Replacement rules: $x \mapsto 1/x$ for the basis elements
`LISTMTX.m`	Replacement rules: $x \mapsto (1-x)/(1+x)$ for the basis elements
`HTransform.nb`	an example notebook for variable transformation
`HPOLY.f`	`FORTRAN` code (1298 lines)
`TTTn.m`	Data files containing the numerical constants needed at the different weights, `n = 2...8`. the different weights
`Bkl.m`	Data files to identify the basis elements: `k=1 .. k=l, k = 5,6,7,8`

Table 5: The list of attached files

The data files `TTTn.m` contain 360, 1600, 5040, 17280, 51040, 162240, 486000 constants for `w` = 2...8. Initially the external files for reading or writing are all set to `STATUS="NEW"`, as e.g.

```
 OPEN(UNIT=14,FILE="TTT2.m",FORM="FORMATTED",STATUS="NEW",
&     ACTION="READ")
```

In later use the standard input files have to be set to `STATUS="OLD"`. For convenience we provide two versions of `HPOLY.f`. `HPOLYnew.f` contains the file status: `"NEW"` and `HPOLY.f` the file status `"OLD"`.

Acknowledgment. We would like to thank R. Germundson, P. Marquard, N. Rana, and K. Schönwald for discussions and N. Rana for an introduction to and help with using `Ginac`. This work was supported in part by the Austrian Science Fund (FWF) grant SFB F50 (F5009-N15), by the bilateral project DNTS-Austria 01/3/2017 (WTZ BG03/2017), funded by the Bulgarian National Science Fund and OeAD (Austria), by 'Innovatives OÖ 2020' of the Upper Austrian Government, by the EU TMR network SAGEX Marie Skłodowska-Curie grant agreement No. 764850 and COST action CA16201: Unraveling new physics at the LHC through the precision frontier. M. Round thanks DESY and the KMP Berlin for support.

References

[1] G. 't Hooft and M.J.G. Veltman, Scalar One Loop Integrals, Nucl. Phys. B **153** (1979) 365–401.

[2] T. Gehrmann and E. Remiddi, Two loop master integrals for $\gamma^* \to 3$ jets: The non-planar topologies, Nucl. Phys. B **601** (2001) 287–317 [hep-ph/0101124].

[3] J.A.M. Vermaseren, A. Vogt and S. Moch, The third-order QCD corrections to deep-inelastic scattering by photon exchange, Nucl. Phys. B **724** (2005) 3–182 [hep-ph/0504242].

[4] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel and C. Schneider, The 3-loop pure singlet heavy flavor contributions to the structure function $F_2(x, Q^2)$ and the anomalous dimension, Nucl. Phys. B **890** (2014) 48–151 [arXiv:1409.1135 [hep-ph]].

[5] E. Remiddi and J.A.M. Vermaseren, Harmonic polylogarithms, Int. J. Mod. Phys. A **15** (2000) 725–754 [hep-ph/9905237].

[6] L. Lewin, Dilogarithms and associated functions, (Macdonald, London, 1958);
L. Lewin, Polylogarithms and associated functions, (North Holland, New York, 1981).

[7] A. Devoto and D.W. Duke, Table of integrals and formulae for Feynman diagram calculations, Riv. Nuovo Cim. **7N6** (1984) 1–39

[8] N. Nielsen, Der Eulersche Dilogarithmus und seine Verallgemeinerungen, Nova Acta Leopold. **XC** (1909) Nr. 3, 125–211.

[9] K.S. Kölbig, J.A. Mignoco and E. Remiddi, On Nielsen's generalized polylogarithms and their numerical calculation, BIT **10** (1970) 38–74.

[10] K.S. Kölbig, Nielsen's generalized polylogarithms, SIAM J. Math. Anal. **17** (1986) 1232–1258.

[11] J. Blümlein, Algebraic relations between harmonic sums and associated quantities, Comput. Phys. Commun. **159** (2004) 19–54 [hep-ph/0311046].

[12] J.A.M. Vermaseren, Harmonic sums, Mellin transforms and integrals, Int. J. Mod. Phys. A **14** (1999) 2037–2076 [hep-ph/9806280].

[13] J. Blümlein and S. Kurth, Harmonic sums and Mellin transforms up to two loop order, Phys. Rev. D **60** (1999) 014018 [hep-ph/9810241].

[14] T. Gehrmann and E. Remiddi, Numerical evaluation of harmonic polylogarithms, Comput. Phys. Commun. **141** (2001) 296–312 [hep-ph/0107173].

[15] C.W. Bauer, A. Frink and R. Kreckel, Introduction to the GiNaC framework for symbolic computation within the C++ programming language, J. Symb. Comput. **33** (2000) 1–12 [cs/0004015 [cs-sc]].

[16] J. Vollinga and S. Weinzierl, Numerical evaluation of multiple polylogarithms, Comput. Phys. Commun. **167** (2005) 177–194 [hep-ph/0410259].

[17] D. Maitre, `HPL`, a mathematica implementation of the harmonic polylogarithms, Comput. Phys. Commun. **174** (2006) 222–240 [hep-ph/0507152]; Extension of `HPL` to complex arguments, Comput. Phys. Commun. **183** (2012) 846 [hep-ph/0703052].

[18] J. Ablinger, The package `HarmonicSums`: computer algebra and analytic aspects of nested sums, PoS (LL2014) 019; A Computer Algebra Toolbox for Harmonic Sums Related to Particle Physics, Diploma Thesis, J. Kepler University Linz, 2009, arXiv:1011.1176 [math-ph].

[19] J. Ablinger, Computer Algebra Algorithms for Special Functions in Particle Physics, Ph.D. Thesis, J. Kepler University Linz, 2012, arXiv:1305.0687 [math-ph].

[20] J. Ablinger, J. Blümlein and C. Schneider, Harmonic sums and polylogarithms generated by cyclotomic polynomials, J. Math. Phys. **52** (2011) 102301 [arXiv:1105.6063 [math-ph]].

[21] J. Ablinger, J. Blümlein and C. Schneider, Analytic and algorithmic aspects of generalized harmonic sums and polylogarithms, J. Math. Phys. **54** (2013) 082301 [arXiv:1302.0378 [math-ph]].

[22] J. Ablinger, J. Blümlein, C.G. Raab and C. Schneider, Iterated binomial sums and their associated iterated integrals, J. Math. Phys. **55** (2014) 112301 [arXiv:1407.1822 [hep-th]].

[23] H. Frellesvig, Generalized polylogarithms in `Maple`, arXiv:1806.02883 [hep-th].

[24] S. Moch, B. Ruijl, T. Ueda, J.A.M. Vermaseren and A. Vogt, Four-loop non-singlet splitting functions in the planar limit and beyond, JHEP **1710** (2017) 041 [arXiv:1707.08315 [hep-ph]].

[25] J. Ablinger, J. Blümlein, P. Marquard, N. Rana and C. Schneider, Heavy quark form factors at three loops in the planar limit, Phys. Lett. B **782** (2018) 528–532 [arXiv:1804.07313 [hep-ph]].

[26] J. Ablinger, J. Blümlein, P. Marquard, N. Rana and C. Schneider, Automated Solution of First Order Factorizable Systems of Differential Equations in One Variable, Nucl. Phys. B, in print, [arXiv:1810.12261 [hep-ph]].

[27] J. Ablinger, J. Blümlein, M. Round and C. Schneider, Special functions, transcendentals and their numerics, PoS (RADCOR 2017) 010 [arXiv:1712.08541 [hep-th]].

[28] J. Blümlein, Analytic continuation of Mellin transforms up to two loop order, Comput. Phys. Commun. **133** (2000) 76–104 [hep-ph/0003100];
J. Blümlein and S.O. Moch, Analytic continuation of the harmonic sums for the 3-loop anomalous dimensions, Phys. Lett. B **614** (2005) 53–61 [hep-ph/0503188];
Structural Relations of harmonic sums and Mellin transforms at weight $w = 6$, Clay Math. Proc. **12** (2010) 167–188 [arXiv:0901.0837 [math-ph]];
A.V. Kotikov and V.N. Velizhanin, Analytic continuation of the Mellin moments of deep inelastic structure functions, hep-ph/0501274.

[29] J. Blümlein, Structural relations of harmonic sums and Mellin transforms up to weight $w = 5$, Comput. Phys. Commun. **180** (2009) 2218–2249 [arXiv:0901.3106 [hep-ph]];

[30] J. Ablinger and J. Blümlein, Harmonic sums, polylogarithms, special numbers, and their generalizations, in : Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions (Texts & Monographs in Symbolic Computation) (Springer, Wien, 2013) 1–29, Eds. C. Schneider and J. Blümlein, arXiv:1304.7071 [math-ph];
J. Ablinger, J. Blümlein and C. Schneider, Generalized harmonic, cyclotomic, and binomial sums, their polylogarithms and special numbers, J. Phys. Conf. Ser. **523** (2014) 012060 [arXiv:1310.5645 [math-ph]].

[31] C. Duhr, Mathematical aspects of scattering amplitudes, in: TASI Lectures Journeys Through the Precision Frontier: Amplitudes for Colliders, Eds. L. Dixon and F. Petriello, p. 419–476, (World Scientific, Singapore, 2015), [arXiv:1411.7538 [hep-ph]].

[32] J. Blümlein and C. Schneider, Analytic computing methods for precision calculations in quantum field theory, Int. J. Mod. Phys. A **33** (2018) no.17, 1830015 [arXiv:1809.02889 [hep-ph]]; Computer algebra tools for Feynman integrals and related multi-sums, PoS (LL2018) 052 [arXiv:1809.06168 [cs.SC]].

[33] J. Blümlein, D.J. Broadhurst and J.A.M. Vermaseren, The multiple zeta value data mine, Comput. Phys. Commun. **181** (2010) 582–625 [arXiv:0907.2557 [math-ph]].

[34] `https://de.wikipedia.org/wiki/Doppelte_Genauigkeit`

[35] R.N. Lee, A.V. Smirnov, V.A. Smirnov and M. Steinhauser, Three-loop massive form factors: complete light-fermion and large-N_c corrections for vector, axial-vector, scalar and pseudo-scalar currents, JHEP **1805** (2018) 187 [arXiv:1804.07310 [hep-ph]]; Three-loop massive form factors: complete light-fermion corrections for the vector current, JHEP **1803** (2018) 136 [arXiv:1801.08151 [hep-ph]];
J. Henn, A.V. Smirnov, V.A. Smirnov and M. Steinhauser, Massive three-loop form factor in the planar limit, JHEP **1701** (2017) 074 [arXiv:1611.07535 [hep-ph]].

[36] J.A.M. Vermaseren, New features of FORM, math-ph/0010025.