

Numerical Implementation of Harmonic Polylogarithms to Weight $w = 8$

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 $w = 8$ at an absolute accuracy of $\sim 4.9 \cdot 10^{-15}$ 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.

1 Introduction

Many analytic higher order and multi-leg calculations result into representations based on harmonic polylogarithms (HPLs) [1–3]. This function space has been introduced in Ref. [4]. It is formed by the Volterra-iterative integrals over the alphabet

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

with

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

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

By definition all HPLs are zero for $x = 0$. The harmonic polylogarithms are generalizations of the classical polylogarithms [5,6] and the Nielsen integrals [7–9]. They obey algebraic [10] and argument relations [4]. Furthermore, they are dual structures, by the Mellin transform

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

to the nested harmonic sums [11,12]

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

In the past numerical representations of a series of classical polylogarithms and Nielsen integrals were given in [8] and have been in wide use for decades. The generalization to harmonic polylogarithms has been performed in [13] up to weight $\mathbf{w} = 4$ and in a private version to $\mathbf{w} = 5$. An implementation using the package `Ginac` [14] has been given in [15]. The `Mathematica` packages `HPL` [16] and `HarmonicSums` [18–22] do also allow the numerical evaluation of harmonic polylogarithms. There is also an implementation in `Maple` [17]. In various applications one needs the numerical representation of higher weight harmonic polylogarithms, e.g. to weight $\mathbf{w} = 6$ in Refs. [23–25], which were given in [26]. Numerical implementations of the analytic continuation of harmonic sums to complex arguments have been given in [27].

In the foreseeable time 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 $\mathbf{w} = 8$. If one compares the computational times between `FORTRAN` implementations and the ones in `Ginac` [15] or `Mathematica` [16,18] 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 numerical implementation for the harmonic polylogarithms. Technical aspects on the code and its use are discussed in Section 3. In Section 4 we comment on the corresponding numerical implementation for cyclotomic harmonic polylogarithms¹ 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.

2 The 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 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. [10], for a 3-letter alphabet

$$N_{\text{bas}}(\mathbf{w}) = \frac{1}{\mathbf{w}} \sum_{d|\mathbf{w}} \mu\left(\frac{\mathbf{w}}{d}\right) 3^d. \quad (6)$$

Here μ denotes the Möbius function.

We have e.g.

$$\text{LIST1.m} = \{H[1, x], H[0, x], H[-1, x]\} \quad (7)$$

$$\text{LIST2.m} = \{H[0, 1, x], H[-1, 1, x], H[-1, 0, x]\} \quad (8)$$

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

¹Beyond these functions various more special functions contribute in analytic calculations in Quantum Field Theories. For recent surveys see [28, 29].

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

$$\text{hrel.m} \quad [12.8 \text{ Mbyte}] \quad (10)$$

attached to this paper.

Usually one has expressions with main argument x of the harmonic polylogarithms in $x \in] - \infty, +\infty[$. For the arguments $x \in] - \infty, 0]$ one first applies the procedure

$$\text{TransformH}[H_{\vec{a}}(f(x)), x] \quad (11)$$

of `HarmonicSums`.

Example:

$$\begin{aligned} \text{TransformH}[H[-1, 0, 0, -x], x] &= -H[1, x]H[0, 0, -x] + H[0, -x]H[0, 1, x] - H[0, 0, 1, x] \\ &= -\frac{1}{2}H[1, x](H[0, x] + i\pi)^2 + (H[0, x] + i\pi) - H[0, 0, 1, x]. \end{aligned} \quad (12)$$

The same command maps $x \in [1, \infty]$ to $x \in [0, 1]$.

Example:

$$\text{TransformH} \left[H \left[-1, 0, 0, \frac{1}{x} \right] \right] = H[-1, 0, 0, x] - \frac{1}{6}H[0, x]^3. \quad (13)$$

Now only two regions have to be considered:

$$x \in [0, \sqrt{2} - 1] \quad \text{and} \quad x \in [\sqrt{2} - 1, 1]. \quad (14)$$

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

$$x \mapsto \frac{1-x}{1+x}. \quad (15)$$

Example:

$$\begin{aligned} \text{TransformH} \left[H \left[-1, 0, 0, \frac{1-x}{1+x} \right] \right] &= \frac{3}{4}\zeta_3 - \frac{1}{6}H[-1, x]^3 - H[-1, x]H[-1, 1, x] \\ &\quad + H[-1, -1, 1, x] - H[-1, 1, 1, x]. \end{aligned} \quad (16)$$

In the case one is leaving the basis one uses

$$\text{ReduceToHBasis}[expression] \quad (17)$$

to get back to the basis again. In order to carry out these function calls, the following subsidiary files are useful. More precisely, the corresponding substitution rules are tabulated in the file `hrel.m`. For the other operations we provide the following substitution files :

$$\text{LISTOMX.m} \quad x \mapsto -x \quad [2.6 \text{ Mbyte}] \quad (18)$$

$$\text{LISTOOX.m} \quad x \mapsto \frac{1}{x} \quad [13.1 \text{ Mbyte}] \quad (19)$$

$$\text{LISTMTX.m} \quad x \mapsto \frac{1-x}{1+x} \quad [13.7 \text{ Mbyte}] \quad (20)$$

In 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

$$\text{HSeparateCuts}[H_{\bar{a}}(x)]. \quad (21)$$

Example:

$$\begin{aligned} \text{HSeparateCuts}[H[-1, 1, 1, x]] &= \{\text{term1}, \text{term2}\} \\ &= \left\{ -\frac{1}{2} \ln^2(2) H[-1, x] + H[-1, 1, 1, x], \frac{1}{2} \ln^2(2) H[-1, x] \right\}. \end{aligned} \quad (22)$$

The sum of the two terms yields the original expression again.

To reduce this representation in optimal terms one also needs the replacement table for the multiple zeta values up to $w=8$ [30]. All the polynomial basis constants contribute,

$$\left\{ \ln(2), \zeta_2, \zeta_3, \text{Li}_4\left(\frac{1}{2}\right), \zeta_5, \text{Li}_5\left(\frac{1}{2}\right), \text{Li}_6\left(\frac{1}{2}\right), s_6, \zeta_7, \text{Li}_7\left(\frac{1}{2}\right), s_{7a}, s_{7b}, \text{Li}_8\left(\frac{1}{2}\right), s_{8a}, s_{8b}, s_{8c}, s_{8d} \right\}, \quad (23)$$

with

$$\zeta_k = S_k(\infty), \quad k \geq 2 \quad (24)$$

$$\text{Li}_k(x) = \sum_{l=1}^{\infty} \frac{x^l}{l^k} \quad (25)$$

$$s_6 = S_{-5,-1}(\infty) \quad (26)$$

$$s_{7a} = S_{-5,1,1}(\infty) \quad (27)$$

$$s_{7b} = S_{5,-1,-1}(\infty) \quad (28)$$

$$s_{8a} = S_{5,3}(\infty) \quad (29)$$

$$s_{8b} = S_{-7,-1}(\infty) \quad (30)$$

$$s_{8c} = S_{-5,-1,-1,-1}(\infty) \quad (31)$$

$$s_{8d} = S_{5,-1,1,1}(\infty). \quad (32)$$

Now the Bernoulli-improvement [31], see also [13], of the respective series expansion is performed. To this end we substitute

$$\text{term1}(\mathbf{x}) \mapsto \text{term1}(\mathbf{x})|_{x \rightarrow 1-e^{-v}} \quad (33)$$

$$\text{term2}(\mathbf{x}) \mapsto \text{term1}(\mathbf{x})|_{x \rightarrow e^u-1}, \quad (34)$$

with

$$u = \ln(1+x) \quad (35)$$

$$v = -\ln(1-x) \quad (36)$$

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 $O(\ln^{w-1}(u))$ and of $O(\ln^{w-2}(v))$. 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 $w = 8$ 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. The corresponding auxiliary files can be used in an usual *Mathematica* session even without invoking the package *HarmonicSums*.

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 0.2375 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 absolute accuracy has been determined comparing with the corresponding result obtained using `Ginac` [15]. 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, seeming to be more subtle in `Mathematica`.

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 :

$$\left\{ \frac{1}{1+x+x^2}, \frac{x}{1+x+x^2}, \frac{1}{1+x^2}, \frac{x}{1+x^2}, \frac{1}{1-x+x^2}, \frac{x}{1-x+x^2} \right\}. \quad (37)$$

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. [24,25]. One possibility is given by Taylor series in x around $x = 0, 1$ and $x = -1$, which can be calculated analytically using the command

$$\text{HarmonicSumsSeries}[\mathbf{f}(x), \{\mathbf{x}, 0, \mathbf{n}\}] \quad (38)$$

of `HarmonicSums`, where \mathbf{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 [33]. The expansion around $x = -1$ may be performed by mapping

$$\left\{ \frac{1}{1+x+x^2} \leftrightarrow \frac{1}{1-x+x^2}, \frac{x}{1+x+x^2} \leftrightarrow -\frac{x}{1-x+x^2} \right\} \quad (39)$$

more effectively. We note that some of the cyclotomic HPLs can be complex for $x \in [-1, 0[$.

x	HPL	run time Fortran [sec]	run time Ginac [sec]	accuracy
-1.0	-0.19456824628601374558e+0	6.0e-6	2.4e-4	5.5e-18
-0.8	-0.10723997730030421271e+0	6.0e-6	2.9e-4	4.5e-18
-0.4	-0.16724871547938951751e-1	8.0e-6	3.3e-4	5.2e-18
0.4	0.25064429005154578702e-1	8.0e-6	2.7e-4	2.3e-18
0.8	0.19221120777067257053e+0	8.0e-6	6.2e-3	6.0e-18
1.0	0.34039020776540250510e+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 accuracy of the respective representation can be tested comparing the numerical result with the complex-valued representations provided by `Ginac`, cf. Ref. [15]. To have these explicit series representations is of importance since they usually perform faster than the algorithm based on a Hölder convolution in [15]. 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.

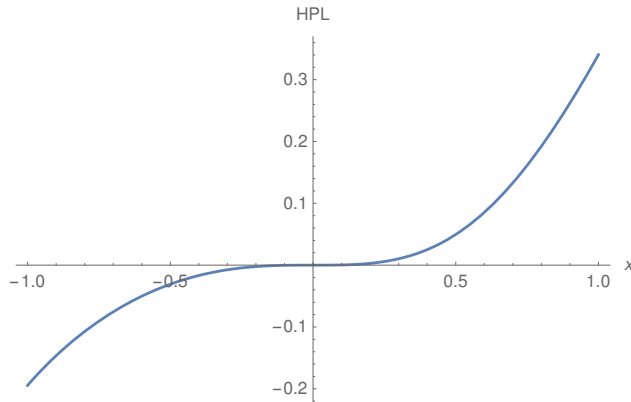


Figure 1: The function $H[\{6,1\},0,-1,x]$ in the region $x \in [-1,1]$.

In case of the planar contributions to the 3-loop massive Formfactors [24, 25] 206 real-valued cyclotomic polylogarithms up to $w=6$ contribute. Their numerical representation is given in Ref. [25].

5 Conclusions

We presented a compact FORTRAN code which calculates the HPLs numerically up to weight $w = 8$. 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 [31]. The numerical accuracy of the representation is $\sim 4.9 \cdot 10^{-15}$ or better. We also compared run times with corresponding implementations in `Ginac`. Because we limit the accuracy to about double precision, the `FORTTRAN` implementation is faster. This is of advantage for larger scale phenomenological and experimental applications. The present code extends earlier work of the authors of Ref. [13] to far higher weight.

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 [24, 25, 32]. 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. [25].

The use of the `FORTTRAN`-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, $H[-1, 1, 0, \mathbf{x}]$ and give some numerical illustration. This harmonic polylogarithm is represented by

$$\begin{aligned}
& H[-1, 1, 0, \mathbf{x}] \\
&= 1.51561421398405852765343970478u - 0.173286795139986327354308030365u^2 \\
&- 9.62704417444368485301711279803e-3u^3 + 4.81352208722184242650855639902e-5u^5 \\
&- 5.45750803539891431576933832088e-7u^7 + 7.95886588495675004383028505128e-9u^9 \\
&- 1.31551502230690083369095620682e-10u^{11} + 2.34789839498327069523925944587e-12u^{13} \\
&- 4.41717498332597960030540850887e-14u^{15} + 8.63804169263087598091731324426e-16u^{17} \\
&- 1.74017681474319135406058068735e-17u^{19} + 3.58929549165142423790521910252e-19u^{21} \\
&- 7.5465410635862794583439948632e-21u^{23} + 1.6120829755323789347574740235e-22u^{25} \\
&- 3.49013136556204594004342294876e-24u^{27} + 7.64298689734656581222853583268e-26u^{29} \\
&- 1.69034929890524597704956962836e-27u^{31} + 3.77081955969303929350628544829e-29u^{33} \\
&- 8.47603948488133928506876824739e-31u^{35} + 1.91812087138166865720209408063e-32u^{37} \\
&- 4.36689003100059767808970725046e-34u^{39} \\
&- 1.51561421398405852765343970478v + 2.45753828564099545590515613688e-1v^2 \\
&- 7.28443369411397782763491496943e-3v^3 - 4.665222471648288365706911026793e-3v^4 \\
&+ 7.18180060085976020176571697142e-4v^5 + 1.0998660373605051065035495605e-5v^6 \\
&- 1.15456136926644459648518129105e-5v^7 + 2.2668351288044216255466542194e-7v^8 \\
&+ 2.10153210137148470728310501335e-7v^9 - 1.14687914794188966052447108046e-8v^{10} \\
&- 3.89908427576215576063242315571e-9v^{11} + 3.66138107063939833413172284743e-10v^{12} \\
&+ 7.06461980894443084509781017833e-11v^{13} - 1.01701961998442644609478487375e-11v^{14} \\
&- 1.20382742279859366919809526002e-12v^{15} + 2.62615442112180864875738033e-13v^{16} \\
&+ 1.80691839540417471780902160508e-14v^{17} - 6.45036155764417783408860256621e-15v^{18}
\end{aligned}$$

$$\begin{aligned}
& -5.34007883918865414818687371719e-12v^{13} + 5.5156739926803903227757686507e-12v^{14} \\
& -6.28857786197662796320841934221e-14v^{15} - 1.18725105791642065839718279224e-13v^{16} \\
& +5.26578132632148224808742741623e-15v^{17} + 2.46845929257087823714218639074e-15v^{18} \\
& -1.99041996903111499913398470405e-16v^{19} - 4.90370649831330005463993519578e-17v^{20} \\
& +6.12701864575634669387293564239e-18v^{21} + 9.12721047200658523982514560198e-19v^{22} \\
& -1.70288180301515490338143682483e-19v^{23} - 1.5276386825065569828307657806e-20v^{24} \\
& +4.42944449090124929492159879193e-21v^{25} + 2.05134350399689853005153453942e-22v^{26} \\
& -1.09492124504286623636232861158e-22v^{27} - 1.10399957724647642108199227876e-24v^{28} \\
& +2.58812712884495900131556798953e-24v^{29} - 6.10104748471770267083048214269e-26v^{30} \\
& -5.85427217667691779189384255622e-26v^{31} + 3.39895981945488747450890358933e-27v^{32} \\
& +1.26193521067501965399016009955e-27v^{33} - 1.22550544038337779976290204537e-28v^{34} \\
& -2.56460584430131563394858221159e-29v^{35} + 3.76155735706687563080727351239e-30v^{36} \\
& +4.80146080059348793629186205266e-31v^{37} - 1.05400577554603083138457524555e-31v^{38} \\
& -7.83689686147728007727555462251e-33v^{39} + 2.77092887621278377430721664171e-33v^{40} \Big] \\
& \times H[0, v] \tag{40}
\end{aligned}$$

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

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

Table 4: Some numerical values for $H[-1, 1, 0, x]$ and the runtimes in the FORTRAN implementation and by Ginac.

The following mappings hold :

$$\begin{aligned}
H[-1, 1, 0, z] &= \frac{1}{2} \ln(2)\zeta_2 - \zeta_3 + \zeta_2 H[-1, x] + \frac{1}{6} H[-1, x]^3 - H[-1, x] H[0, -1, x] \\
&\quad - H[-1, x] H[0, 1, x] + H[-1, -1, 1, x] + 2H[0, -1, -1, x] + H[0, -1, 1, x] \\
&\quad + H[0, 1, -1, x] \text{ for } z \in [\sqrt{2} - 1, 1], \quad z = \frac{1-x}{1+x}. \tag{41}
\end{aligned}$$

$$H[-1, 1, 0, -x] = H[-1, x]H[0, -x]H[1, x] - H[0, -x]H[-1, 1, x] - H[1, x]H[0, -1, x] + H[0, -1, 1, x] \quad (42)$$

$$H\left[-1, 1, 0, \frac{1}{x}\right] = 3 \ln(2)\zeta_2 + \frac{3}{4}\zeta_3 - 2\zeta_2 H[-1, x] + 2\zeta_2 H[0, x] - \frac{1}{2}H[-1, x]H[0, x]^2 + \frac{1}{6}H[0, x]^3 - H[0, x]H[-1, 1, x] + H[0, x]H[0, -1, x] + H[-1, x]H[0, 1, x] + H[0, x]H[0, 1, x] - H[0, 0, -1, x] - 2H[0, 0, 1, x] - H[0, 1, -1, x]. \quad (43)$$

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 runtimes of **HPOLY** and **Ginac** [15]. 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.

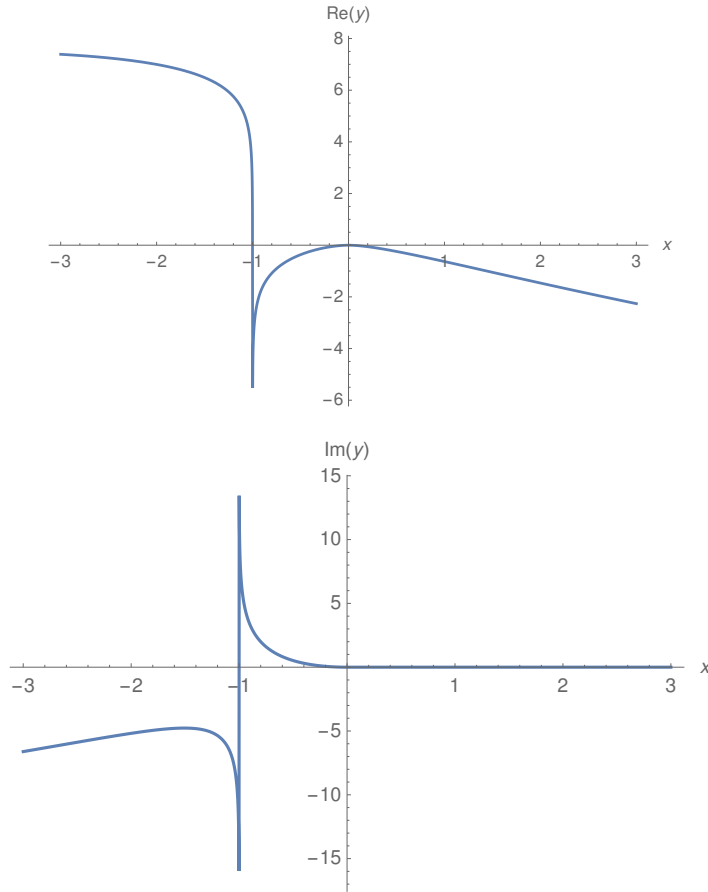


Figure 2: Real and imaginary part of the function $H[-1, 1, 0, x]$.

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

$$\mathbb{H}[\{6, 1\}, 0, -1, \mathbf{x}] = \int_0^x dy_1 \frac{y_1}{1 - y_1 + y_1^2} \int_0^{y_1} \frac{dy_2}{y_2} \int_0^{y_2} \frac{dy_2}{1 + y_2}, \quad (44)$$

$$\begin{aligned}
& +0.276765717942188530423824541472e-3x_+^{17} + 0.232655869910771871556185281675e-3x_+^{18} \\
& +0.195199537304800462695199537305e-3x_+^{19} + 0.164375742936114453142316919407e-3x_+^{20} \\
& +0.139449719281652054761298458777e-3x_+^{21} + 0.119375300560214717265097225694e-3x_+^{22} \\
& +0.131134477762556911836761076230e-3x_+^{23} + 0.897896648502382786943293067940e-4x_+^{24} \\
& +0.787312103101576785787312103102e-4x_+^{25} + 0.694430190195227419411231291907e-4x_+^{26} \\
& +0.615646514102514860305899426862e-4x_+^{27} + 0.548299963598119473742033978943e-4x_+^{28} \\
& +0.490370738953877725049414746362e-4x_+^{29} + 0.440279180491117866083633973683e-4x_+^{30} \\
& +0.396759437506466368945146534281e-4x_+^{31} + 0.358780629639696323771643547090e-4x_+^{32} \\
& +0.325495384282835477024478780285e-4x_+^{33} + 0.296203528090464701585299046187e-4x_+^{34} \\
& +0.270324458622603140106269524853e-4x_+^{35} + 0.247375088378105046141745858401e-4x_+^{36} \\
& +0.226951904770740284572302948414e-4x_+^{37} + 0.208716366275858749420441086658e-4x_+^{38} \\
& +0.192383083866313435144166242061e-4x_+^{39} + 0.177710310170514453671986061289e-4x_+^{40} \\
& \times H[-1, x] \tag{47}
\end{aligned}$$

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** [34] 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
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
Bk1.m	Data files to identify the basis elements: k=1 .. k=1, 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 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] T. Gehrmann and E. Remiddi, Two loop master integrals for $\gamma^* \rightarrow 3$ jets: The non-planar topologies, Nucl. Phys. B **601** (2001) 287 [hep-ph/0101124].
- [2] 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 [hep-ph/0504242].
- [3] 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 [arXiv:1409.1135 [hep-ph]].
- [4] E. Remiddi and J.A.M. Vermaseren, Harmonic polylogarithms, Int. J. Mod. Phys. A **15** (2000) 725 [hep-ph/9905237].
- [5] L. Lewin, Dilogarithms and associated functions, (Macdonald, London, 1958);
L. Lewin, Polylogarithms and associated functions, (North Holland, New York, 1981).
- [6] A. Devoto and D.W. Duke, Table of integrals and formulae for Feynman diagram calculations, Riv. Nuovo Cim. **7N6** (1984) 1.
- [7] N. Nielsen, Der Eulersche Dilogarithmus und seine Verallgemeinerungen, Nova Acta Leopold. **XC** (1909) Nr. 3, 125.
- [8] K.S. Kölbig, J.A. Mignoco and E. Remiddi, On Nielsen's generalized polylogarithms and their numerical calculation, BIT **10** (1970) 38.
- [9] K.S. Kölbig, Nielsen's generalized polylogarithms, SIAM J. Math. Anal. **17** (1986) 1232.
- [10] J. Blümlein, Algebraic relations between harmonic sums and associated quantities, Comput. Phys. Commun. **159** (2004) 19 [hep-ph/0311046].
- [11] J.A.M. Vermaseren, Harmonic sums, Mellin transforms and integrals, Int. J. Mod. Phys. A **14** (1999) 2037 [hep-ph/9806280].

- [12] J. Blümlein and S. Kurth, Harmonic sums and Mellin transforms up to two loop order, *Phys. Rev. D* **60** (1999) 014018 [hep-ph/9810241].
- [13] T. Gehrmann and E. Remiddi, Numerical evaluation of harmonic polylogarithms, *Comput. Phys. Commun.* **141** (2001) 296 [hep-ph/0107173].
- [14] 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 [cs/0004015 [cs-sc]].
- [15] J. Vollinga and S. Weinzierl, Numerical evaluation of multiple polylogarithms, *Comput. Phys. Commun.* **167** (2005) 177 [hep-ph/0410259].
- [16] D. Maitre, HPL, a mathematica implementation of the harmonic polylogarithms, *Comput. Phys. Commun.* **174** (2006) 222 [hep-ph/0507152]; Extension of HPL to complex arguments, *Comput. Phys. Commun.* **183** (2012) 846 [hep-ph/0703052].
- [17] H. Frellesvig, Generalized polylogarithms in Maple, arXiv:1806.02883 [hep-th].
- [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] 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]].
- [24] 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 [arXiv:1804.07313 [hep-ph]].
- [25] J. Ablinger, J. Blümlein, P. Marquard, N. Rana and C. Schneider, DESY 18-053, DO-TH 18/09.
- [26] J. Ablinger, J. Blümlein, M. Round and C. Schneider, Special functions, transcendentals and their numerics, PoS (RADCOR 2017) 010 [arXiv:1712.08541 [hep-th]].
- [27] J. Blümlein, Analytic continuation of Mellin transforms up to two loop order, *Comput. Phys. Commun.* **133** (2000) 76 [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 [hep-ph/0503188];
 J. Blümlein, Structural relations of harmonic sums and Mellin transforms up to weight $w = 5$, *Comput. Phys. Commun.* **180** (2009) 2218; [arXiv:0901.3106 [hep-ph]]; Structural Relations of harmonic sums and Mellin transforms at weight $w = 6$, *Clay Math. Proc.* **12** (2010) 167 [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.
- [28] 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) pp. 1, 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]].
- [29] J. Blümlein and C. Schneider, *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]].
- [30] J. Blümlein, D.J. Broadhurst and J.A.M. Vermaseren, The multiple zeta value data mine, *Comput. Phys. Commun.* **181** (2010) 582 [arXiv:0907.2557 [math-ph]].
- [31] G. 't Hooft and M.J.G. Veltman, Scalar One Loop Integrals, *Nucl. Phys. B* **153** (1979) 365.
- [32] 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]].
- [33] https://de.wikipedia.org/wiki/Doppelte_Genauigkeit
- [34] J.A.M. Vermaseren, New features of FORM, math-ph/0010025.