SFB/CPP-13-50
TTP13-27
LPN13-052

arXiv:1307.6925v1 [hep-ph] 26 Jul 2013

MT: a Mathematica package to compute convolutions

Maik Höschele[a], Jens Hoff[a], Alexey Pak[a], Matthias Steinhauser[a,*], Takahiro Ueda[a]

[a] *Institut für Theoretische Teilchenphysik, Karlsruhe Institute of Technology (KIT), 76128 Karlsruhe, Germany*

Abstract

We introduce the Mathematica package MT which can be used to compute, both analytically and numerically, convolutions involving harmonic polylogarithms, polynomials or generalized functions. As applications contributions to next-to-next-to-next-to leading order Higgs boson production and the Drell-Yan process are discussed.

*Corresponding author

Email addresses: maik.hoeschele@kit.edu (Maik Höschele), jens.hoff@kit.edu (Jens Hoff), alexey.pak@iosb.fraunhofer.de (Alexey Pak), matthias.steinhauser@kit.edu (Matthias Steinhauser), takahiro.ueda@kit.edu (Takahiro Ueda)

Program summary

Title of program: `MT`

Available from:
`http://www-ttp.physik.uni-karlsruhe.de/Progdata/ttp13/ttp13-27/`

Computer for which the program is designed and others on which it is operable: Any computer where `Mathematica` version 6 or higher is running.

Operating system or monitor under which the program has been tested: Linux

No. of bytes in distributed program including test data etc.: approximately 50 000 bytes, and tables of approximately 60 megabytes

Distribution format: source code

Keywords: Convolution of partonic cross sections and splitting functions, Mellin transformation, harmonic sums, harmonic polylogarithms, Higgs boson production, Drell-Yan process

Nature of physical problem: For the treatment of collinear divergences connected to initial-state radiation it is necessary to consider convolutions of partonic cross sections with splitting functions. `MT` can be used to compute such convolutions.

Method of solution: `MT` is implemented in `Mathematica` and we provide several functions in order to perform transformations to Mellin space, manipulations of the expressions, and inverse Mellin transformations.

Restrictions on the complexity of the problem: In case the weight of the input quantities is too high the tables for the (inverse) Mellin transforms have to be extended. In the current implementation the tables contain expressions up to weight eight, code for the generation of tables of even higher weight is provided, too.
`MT` can only handle convolutions of expressions involving harmonic polylogarithms, plus distributions and polynomials in the partonic variable x.

Typical running time: In general the run time for the individual operations is at most of the order of a few minutes (depending on the speed and memory of the computer).

1. Introduction

The calculation of higher order perturbative corrections within QCD inevitably contains contributions where quarks and/or gluons are radiated off initial-state partons. Such processes are accompanied by infra-red singularities which arise from collinear emissions.

These divergences must cancel against convolutions of splitting functions with lower-order cross sections which provide subtraction terms leading to finite partonic cross sections. The origin of the subtraction terms are ultra-violet divergences present in the bare parton densities which, due to the masslessness of all involved partons, can be viewed as counterterms for collinear divergences in the partonic cross section [1]. Alternatively, one can consider the removal of the divergences as a redefinition of the parton distribution functions (PDFs). Since the latter are in general given in the $\overline{\text{MS}}$ scheme we adopt this scheme also for our calculations.

In this paper we provide a description of the `Mathematica` package `MT` which can be used to perform convolutions of splitting functions with partonic cross sections expressed in terms of harmonic polylogarithms (HPLs), delta functions, plus distributions, polynomials in x, and factors $1/x$, $1/(1-x)$, and

$1/(1+x)$. Here x is the partonic variable typically defined as $x = M^2/s$ where M is the mass of the (intermediate) final state and $\sqrt{s}$ is the partonic center-of-mass energy.

The method to compute convolutions implemented in `MT` relies heavily on the Mellin transformation and its properties. For a comprehensive discussion of the Mellin transform and the list of all Mellin images appearing in the calculation of the next-to-next-to-leading order (NNLO) Higgs boson production rate see Refs. [2, 3]. In contrast, we decided to relate all required results to a limited set of Mellin transforms of HPLs with a certain maximum weight.

As applications we discuss convolution contributions to next-to-next-to-next-to-leading order (N^{3}LO) Higgs boson production and the Drell-Yan process. In particular we provide the partonic cross sections up to NNLO expanded sufficiently deep in $\epsilon = (4-D)/2$, where D is the number of space-time dimensions. As far as Higgs boson production is concerned the results have been obtained in Refs. [4, 5]. The corresponding results for the Drell-Yan process are new. It is convenient to introduce the following notation for the partonic cross sections

$$\tilde{\sigma}_{ij}(x) = A\left[\tilde{\sigma}_{ij}^{(0)}(x) + \frac{\alpha_s}{\pi}\tilde{\sigma}_{ij}^{(1)}(x) + \left(\frac{\alpha_s}{\pi}\right)^2 \tilde{\sigma}_{ij}^{(2)}(x) + \ldots\right], \quad (1)$$

where A collects all constants such that

$$\tilde{\sigma}_{ij}^{(0),\mathrm{Higgs}} = \delta_{ig}\delta_{jg}\frac{\delta(1-x)}{1-\epsilon}, \quad (2)$$

and

$$\tilde{\sigma}_{ij}^{(0),\mathrm{DY}} = \delta_{iq}\delta_{j\bar{q}}\delta(1-x)(1-\epsilon). \quad (3)$$

The indices i and j in Eq. (1) refer to the partons in the initial state, i.e., gluons and massless quarks.

The crucial input to the convolution integrals are the splitting functions $P_{ij}(x)$ describing the probability of parton j to turn by emission into parton i with the fraction x of its initial energy. We define the perturbative expansion by

$$P_{ij}(x) = \delta_{ij}\delta(1-x) + \frac{\alpha_s}{\pi}P_{ij}^{(1)}(x) + \left(\frac{\alpha_s}{\pi}\right)^2 P_{ij}^{(2)}(x) + \left(\frac{\alpha_s}{\pi}\right)^3 P_{ij}^{(3)}(x) + \ldots, \quad (4)$$

where the analytic results for the k-loop splitting function $P_{ij}^{(k)}(x)$ can be found in Refs. [6, 7, 8, 9]. Together with the program `MT` we provide a `Mathematica` file `splitfnsMVV.m` which contains splitting functions computed in [8, 9] in a slightly modified form. Namely, the upper index is shifted by one and the singular contributions are stated explicitly in terms of generalized functions. In Tab. 1 we provide the translation from the notation used in the file to that of Eq. (4). Note that for the quark contributions the indices i and j are omitted and additional super- and subscripts are introduced in order to distinguish the various contributions.

The outline of the paper is as follows: the algorithm realized in `MT` is described in the following section, its functionality in Section 3. Examples for the application of `MT` are given in Sections 4 and 5 where the subtraction terms are discussed for Higgs boson production and the Drell-Yan process at N^{3}LO. A summary is given in Section 6.

2. Systematic approach to convolution integrals

Convolution integrals of partonic cross sections and splitting functions are defined as

$$[f \otimes g](x) = \int_0^1 \mathrm{d}x_1 \mathrm{d}x_2 \delta(x - x_1 x_2) f(x_1) g(x_2), \quad (5)$$

splitfnsMVV.m	Eq. (4)	comments, connection to [8, 9]
Pgg1	$P^{(1)}_{gg}$	—
Pgq1	$P^{(1)}_{gq}$	—
Pqg1	$P^{(1)}_{qg}$	—
Pqq1	$P^{(1)}_{\rm ns}$	—
Pgg2	$P^{(2)}_{gg}$	—
Pgq2	$P^{(2)}_{gq}$	—
Pqg2	$P^{(2)}_{qg}$	—
Pnsp2	$P^{(2)+}_{\rm ns}$	quark-quark non-singlet, equals $P^{\rm v}_{qq} + P^{\rm v}_{q\bar{q}}$
Pnsm2	$P^{(2)-}_{\rm ns}$	quark-quark non-singlet, equals $P^{\rm v}_{qq} - P^{\rm v}_{q\bar{q}}$
Pps2	$P^{(2)}_{\rm ps}$	quark-quark pure singlet
Pgg3	$P^{(3)}_{gg}$	—
Pgq3	$P^{(3)}_{gq}$	—
Pqg3	$P^{(3)}_{qg}$	—
Pnsp3	$P^{(3)+}_{\rm ns}$	quark-quark non-singlet, equals $P^{\rm v}_{qq} + P^{\rm v}_{q\bar{q}}$
Pnsm3	$P^{(3)-}_{\rm ns} + P^{(3){\rm s}}_{\rm ns}$	quark-quark non-singlet, equals $P^{\rm v}_{qq} - P^{\rm v}_{q\bar{q}} + P^{\rm s}_{\rm ns}$,
		$P^{(3){\rm s}}_{\rm ns}$ is proportional to $d^{abc}d_{abc}/n_c$
Pps3	$P^{(3)}_{\rm ps}$	quark-quark pure singlet

Table 1: Notation for $P^{(k)}_{ij}$ used in splitfnsMVV.m and in Eq. (4). The superscripts v and s stand for "valence" and "sea". Note that the superscript indicating the loop-order is shifted in our notation compared to Refs. [8, 9] by +1 and that $P^{(3){\rm s}}_{\rm ns}$ appears at three loops for the first time.

where the functions f and g include combinations of HPLs (for the definition see below) up to certain maximum weight with factors $1/x$, $1/(1-x)$, and $1/(1+x)$, polynomials in x, and the generalized functions $\delta(1-x)$ and $\left[\frac{\ln^k(1-x)}{1-x}\right]_+$.

Although the algorithm used for the computation of the convolution integrals has already been presented before (see Appendix B of [10] and Ref. [4]), we decided to review its description for completeness' sake. An alternative way to compute the same convolution integrals has been presented in Ref. [5].

Convolutions $[f \otimes g]$ are most conveniently dealt with by considering Mellin transforms defined through

$$M_n\left[f(x)\right] \;=\; \int_0^1 {\rm d}x x^{n-1} f(x)\,, \tag{6}$$

since in Mellin space convolution integrals turn into products of Mellin images:

$$M_n\big[[f\otimes g](x)\big] \;=\; M_n\left[f(x)\right] M_n\left[g(x)\right]\,. \tag{7}$$

Mellin transforms relate HPLs and their derivatives to harmonic sums [11, 12]. HPLs $\mathrm{H}_{\vec{a}}(x)$ are nested integrals, recursively defined via

$$\mathrm{H}(x)=1,\;\; \mathrm{H}_{a,\vec{b}}(x) = \int_0^x {\rm d}x' f_a(x')\; \mathrm{H}_{\vec{b}}(x'),\;\;\; f_0(x)=\frac{1}{x},\;\; f_1(x)=\frac{1}{1-x},\;\; f_{-1}(x)=\frac{1}{1+x}\,, \tag{8}$$

where one refers to the number of indices as weight. Up to weight three HPLs may be represented by combinations of ordinary logarithms and Nielsen polylogarithms. E.g. up to weight one we have

$$\begin{aligned} \mathrm{H}(x) &= 1\,, \\ \mathrm{H}_0(x) &= \ln x\,, \\ \mathrm{H}_1(x) &= -\ln(1-x)\,, \\ \mathrm{H}_{-1}(x) &= \ln(1+x)\,. \end{aligned} \tag{9}$$

Harmonic sums $\mathrm{S}_{\vec{a}}(x)$ are defined in a similar fashion as nested sums:

$$\mathrm{S}(n) = 1, \quad \mathrm{S}_{a,\vec{b}}(n) = \sum_{i=1}^{n} f_a(i)\ \mathrm{S}_{\vec{b}}(i), \quad f_a(i) = \begin{cases} i^{-a}, & a \geq 0\,, \\ (-1)^i\ i^{a}, & a < 0\,, \end{cases} \tag{10}$$

where the weight is defined as the sum of the absolute values of the indices.

For illustration we demonstrate Mellin transforms of HPLs through weight one:

$$\begin{aligned} M_n[1] &= \frac{1}{n}\,, \\ M_n[\mathrm{H}_0(x)] &= -\frac{1}{n^2}\,, \\ M_n[\mathrm{H}_1(x)] &= \frac{\mathrm{S}_1(n)}{n}\,, \\ M_n[\mathrm{H}_{-1}(x)] &= -\frac{(-1)^n}{n}\,(\mathrm{S}_{-1}(n) + \ln 2) + \frac{\ln 2}{n}\,. \end{aligned} \tag{11}$$

The main ingredient in our algorithm are relations among Mellin transforms of functions $f(x)$, $x^k f(x)$ and $df(x)/dx$ established with the help of integration-by-parts identities. From the definition of the Mellin transform it is obvious that

$$M_n\left[x^k f(x)\right] = M_{n+k}\left[f(x)\right]\,, \tag{12}$$

and furthermore if $f(x)$ is regular for $x \to 1$, then

$$M_n\left[\frac{d}{dx} f(x)\right] = x^{n-1} f(x)\Big|_0^1 - (n-1) M_{n-1}\left[f(x)\right]\,. \tag{13}$$

Note that in the limit $x \to 0$ the boundary term in Eq. (13) vanishes since we always consider n higher than the order of the highest pole that $f(x)$ may have at $x = 0$. However, in the limit $x \to 1$ logarithmic singularities of the form $\ln^k(1-x)$ may appear. They are treated with the help of "regularized derivatives" $\hat{\partial}_x$ which are defined as

$$\begin{aligned} M_n\left[\hat{\partial}_x 1\right] &= 1\,, \\ M_n\left[\hat{\partial}_x f(x)\right] &= R\left[f(x)\right] - (n-1) M_{n-1}\left[f(x)\right]\,. \end{aligned} \tag{14}$$

The operation R regulates the boundary term by dropping all logarithmically divergent contributions:

$$R[g_k(x)\ln^k(1-x) + g_{k-1}(x)\ln^{k-1}(1-x) + \ldots + g_0(x)] = g_0(1)\,, \quad g_k(1) \neq 0 \quad \forall k > 0\,. \tag{15}$$

To motivate this definition let us remark that it corresponds to regulating divergences by means of delta and plus distributions. As an example consider $f(x) = \mathrm{H}_1(x) = -\ln(1-x)$. Since $d\mathrm{H}_1(x)/dx = 1/(1-x)$ the Mellin transform $M_n[df(x)/dx]$ is not defined. On the other hand, if we replace $1/(1-x)$ by the corresponding plus distribution we have

$$M_n\left[\left[\frac{1}{1-x}\right]_+\right] = -\mathrm{S}_1(n-1)\,. \tag{16}$$

The same result is obtained by applying Eq. (14) as can be seen by the following chain of equations

$$M_n\left[\hat{\partial}_x \mathrm{H}_1(x)\right] = R\left[-\ln(1-x)\right] - (n-1)M_{n-1}\left[-\ln(1-x)\right] = 0 - (n-1)\frac{\mathrm{S}_1(n-1)}{n-1} = -\mathrm{S}_1(n-1)\,.$$

Furthermore, in case $f(x)$ is not divergent for $x \to 1$ Eq. (14) reduces to the usual derivative (see Eq. (13)). For example,

$$\hat{\partial}_x \mathrm{H}_{-1}(x) = \frac{d}{dx}\mathrm{H}_{-1}(x) = \frac{1}{1+x}\,,\quad \hat{\partial}_x \mathrm{H}_{-1,1}(x) = \frac{d}{dx}\mathrm{H}_{-1,1}(x) = \frac{\mathrm{H}_1(x)}{1+x}\,. \tag{17}$$

The concept of regularized derivatives relates HPLs to "common" generalized functions. This allows for a unified treatment of Mellin transforms of derivatives of HPLs, independent of the presence of divergences. Hence it is sufficient to consider Mellin transforms of regularized derivatives in case Mellin transforms for generalized functions are needed. To illustrate this point let us show a few examples for regularized derivatives of HPLs

$$\begin{aligned}
\hat{\partial}_x\, 1 &= \delta(1-x)\,,\\
\hat{\partial}_x \mathrm{H}_1(x) &= \left[\frac{1}{1-x}\right]_+\,,\\
\hat{\partial}_x \mathrm{H}_{1,1}(x) &= -\left[\frac{\ln(1-x)}{1-x}\right]_+\,,\\
\hat{\partial}_x \mathrm{H}_{1,1,1}(x) &= \frac{1}{2}\left[\frac{\ln^2(1-x)}{1-x}\right]_+\,,\\
\hat{\partial}_x \mathrm{H}_{1,2}(x) &= \frac{\pi^2}{6}\left[\frac{1}{1-x}\right]_+ + \frac{\mathrm{H}_2(x)-\frac{\pi^2}{6}}{1-x}\,.
\end{aligned} \tag{18}$$

To continue the example of HPLs up to weight one, we list the Mellin transforms of their regularized derivatives

$$\begin{aligned}
M_n[\hat{\partial}_x 1] &= 1\,,\\
M_n[\hat{\partial}_x \mathrm{H}_0(x)] &= \frac{1}{n-1}\,,\\
M_n[\hat{\partial}_x \mathrm{H}_1(x)] &= -\mathrm{S}_1(n-1)\,,\\
M_n[\hat{\partial}_x \mathrm{H}_{-1}(x)] &= (-1)^{n-1}\mathrm{S}_{-1}(n-1) + (-1)^{n-1}\ln 2\,.
\end{aligned} \tag{19}$$

One can interpret the combination of Mellin transforms of HPLs (see Eq. (11)) and their generalized derivatives (see Eq. (19)) computed up to a fixed maximum weight as a system of linear equations. This system can be solved for monomials of the form $1/n^k$, $\mathrm{S}_{\vec{a}}(n)/n^k$, and $(-1)^n\mathrm{S}_{\vec{a}}(n)/n^k$. The solution is then equivalent to the inverse Mellin transform.

We conclude our example for quantities up to weight one by providing their inverse Mellin transforms:

$$\begin{aligned}
M_x^{-1}\left[\frac{1}{n}\right] &= 1\,,\\
M_x^{-1}\left[1\right] &= \hat{\partial}_x 1\,,\\
M_x^{-1}\left[\frac{1}{n^2}\right] &= -\mathrm{H}_0(x)\,,\\
M_x^{-1}\left[S_1(n)\right] &= -x\hat{\partial}_x \mathrm{H}_1(x)\,,\\
M_x^{-1}\left[\frac{S_1(n)}{n}\right] &= \mathrm{H}_1(x)\,.
\end{aligned} \tag{20}$$

The discussion above allows us to formulate the following algorithm. Given the task of finding a convolution of functions f and g, one may perform the following steps.

1. Transform the expressions f and g to Mellin n-space.
2. Compute a table of Mellin transforms of HPLs up to a fixed maximum weight.
3. Prepare a corresponding table holding regularized derivatives of HPLs.
4. Solve the system of linear equations composed of the tables from steps 2 and 3.
5. Perform inverse transformation to x-space by substituting the results from step 4 into the expression $M_n^{-1}\left[M_n\left[f\right]M_n\left[g\right]\right]$.

In step 2, Mellin transforms of HPLs are obtained by using the `FORM` package `harmpol` [13]. In step 3, Mellin transforms of regularized derivatives of HPLs are obtained from the result of step 2 via Eq. (14). Although some inverse transforms cannot be determined from the system of equations in step 4, in practice they cancel in final expressions for the convolutions in step 5. In `MT` the precomputed results of steps 2 and 4 are tabulated to speed-up the calculation and provided in the form of table files.

In the considered examples of Higgs boson and vector boson production (cf. Sections 4 and 5) we were able to compute all necessary convolutions relevant at N^{3}LO applying this algorithm.

3. Description of `MT`

`MT` is a `Mathematica` package for computing analytically convolutions of HPLs with factors $1/x$, $1/(1-x)$ and $1/(1+x)$, polynomials in x, and generalized functions $\delta(1-x)$ and $\left[\frac{\ln^k(1-x)}{1-x}\right]_+$, via Mellin transforms. For cross checks it also provides capability to compute convolutions numerically. It requires `Mathematica` version 6 or later and the `HPL` package [14, 15]. Having installed `MT` properly as described in the provided `README` file, one can load it via

```
In[1]    :=  <<MT`
```

The package handles the following objects:

- `HPL[{`m_1`,`$\ldots$`,`m_N`}, `x`]` $= \mathrm{H}_{m_1,\ldots,m_N}(x)$ is the HPL provided by `HPL` package.
- `PlusDistribution[`k`, `$1-x$`]` $= \left[\frac{\ln^k(1-x)}{1-x}\right]_+$ is the plus distribution ($k = 0, 1, 2, \ldots$).
  `PlusDistrubition[`-1`, `$1-x$`]` $= \delta(1-x)$ is the delta function.
- `DReg[`f`, `x`]` $= \hat{\partial}_x f$ is the regularized derivative of the expression f with respect to x which can be obtained by replacing every singular term $\frac{\ln^k(1-x)}{1-x}$ in the result of the ordinary differentiation $\frac{df}{dx}$ with a plus distribution $\left[\frac{\ln^k(1-x)}{1-x}\right]_+$. We define `DReg[1, `$x$`]` $= \hat{\partial}_x 1 = \delta(1-x)$. Mellin transforms of regularized derivatives are given by Eq. (14).
- `HSum[{`m_1`,`$\ldots$`,`m_N`}, `n`]` $= \mathrm{S}_{m_1,\ldots,m_N}(n)$ is the harmonic sum which appears in Mellin transforms of HPLs.

For computing convolutions, both analytically and numerically, `MT` package provides the following functions:

- `Convolution[`f_1`,...,`f_N`,`x`]` gives the convolution with respect to x of the expressions f_1, ..., f_N,

$$[f_1 \otimes \ldots \otimes f_N](x) = \int_0^1 dx_1 \ldots \int_0^1 dx_N\, f_1(x_1) \ldots f_N(x_N) \delta(x - x_1 \ldots x_N). \tag{21}$$

For illustration we consider the following examples:

```
In[2]  := Convolution[HPL[{0},x]/(1+x),HPL[{0},x]/(1-x),x]
```

Out[2] = $\frac{\mathrm{H}_{-2,0}(x)}{x+1} - \frac{\mathrm{H}_{2,0}(x)}{x+1} - \frac{\mathrm{H}_{0,0,0}(x)}{x+1} - \frac{\pi^2\,\mathrm{H}_0(x)}{12(x+1)} - \frac{\zeta(3)}{2(x+1)}$

```
In[3]  := Convolution[PlusDistribution[0,1-x],(HPL[{2},x]-HPL[{2},1])/(1-x),x]
```

Out[3] = $-\frac{\mathrm{H}_{1,2}(x)}{1-x} - \frac{\mathrm{H}_{2,0}(x)}{1-x} - \frac{2\,\mathrm{H}_{2,1}(x)}{1-x} + \frac{\pi^2\,\mathrm{H}_0(x)}{6(1-x)} + \frac{\pi^2\,\mathrm{H}_1(x)}{6(1-x)} - \frac{\mathrm{H}_3(x)}{1-x} - \frac{\zeta(3)}{1-x}$

```
In[4]  := Convolution[PlusDistribution[0,1-x],PlusDistribution[0,1-x],x]
```

Out[4] = $2\left[\frac{\log(1-x)}{1-x}\right]_+ - \frac{\mathrm{H}_0(x)}{1-x} - \frac{1}{6}\pi^2\delta(1-x)$

Since `Convolution` converts the convolution into a product of Mellin transforms of the expressions and then performs the inverse Mellin transform to obtain the result, the maximum weight of the expressions that `Convolution` can handle is limited by the weight implemented in the tables of the transforms. In the current version `MT` can handle expressions up to weight six by default,[1] tables up to weight eight can be loaded if necessary. In case higher weights are needed `Convolution` returns a result containing unevaluated Mellin and inverse Mellin transforms.

- `NConvolution[`f_1`,...,`f_N`,`x`,`a`]` numerically computes the convolution with respect to x of the expressions f_1, ..., f_N at $x = a$, by using `Mathematica`'s built-in `NIntegrate` function. In case the result contains plus distributions `NConvolution` evaluates the corresponding coefficients numerically. The numerical computation of the above examples for $x = 3/10$ looks as follows:

```
In[5]  := NConvolution[HPL[{0},x]/(1+x),HPL[{0},x]/(1-x),x,3/10]
```

Out[5] = 0.600799

```
In[6]  := NConvolution[PlusDistribution[0,1-x],(HPL[{2},x]-HPL[{2},1])/(1-x),x,3/10]
```

Out[6] = $-2.86725 + 1.7975 \times 10^{-17} i$

```
In[7]  := NConvolution[PlusDistribution[0,1-x],PlusDistribution[0,1-x],x,3/10]
```

Out[7] = $2.\left[\frac{\log(1-x)}{1-x}\right]_+ - 1.64493\,\delta(1-x) + 1.71996$

Currently `NConvolution` supports convolutions of up to three functions exclusive of delta functions, which can be removed trivially. If the expressions contain plus distributions the auxiliary

[1] In the final result for the N^3LO contributions only weight-five HPLs appear. However, in intermediate steps harmonic sums of weight six are present which require the corresponding tables.

regularization [16]

$$\frac{\ln^k(1-x)}{1-x} \to \lim_{a\to 1} \frac{1}{\eta^k} \frac{\partial^k}{\partial a^k} (1-x)^{-1+a\eta} \tag{22}$$

is introduced before the numerical integration of each individual term. In a next step the singularities in the integrals for $x \to 0$ and $x \to 1$ are separated, the differentiation w.r.t. a is performed, the limit $a \to 1$ is taken, and the result is expanded in η. All pole terms $1/\eta^k$ should cancel among the integrals and the leading term gives the result of the convolution. The remaining integrals must have no divergences and can be performed numerically. `NConvolution` checks that all pole terms cancel within an absolute tolerance controlled by `Tolerance` option (10^{-6} by default).

Since `NConvolution` performs the convolution numerically, the result must be finite. Moreover it must be possible to evaluate the integrand numerically, i.e., it cannot contain any symbols, except those specified by `Constants` option. If `Constants` option is used, `NConvolution` tries to collect terms involving the same powers of these constants and splits the convolution with respect to them. The use of `Constants` is illustrated by the following example:

```
In[8]   :=  NConvolution[(1+c)*x+x^2,HPL[{0},x],x,3/10,Constants->{c}]
```

Out[8] = $-0.503973\,c - 0.878459$

`NConvolution` accepts also the following options, which are passed to `NIntegrate`: `AccuracyGoal`, `MaxPoints`, `MaxRecursion`, `Method`, `MinRecursion`, `PrecisionGoal`, and `WorkingPrecision`. One may need to change these options when the convergence of the numerical integration is slow. Note that `NConvolution` may turn out to be very slow if it is called with more than two functions in the argument.

- `NEval[`f`, `x`, `a`]` gives the numerical value of the expression f at $x = a$ and is similar to `N[`f` /. `$x \to a$`]` but does not touch x in generalized functions. This is convenient in case one wants to obtain a numerical value by substituting a certain value of x into a result of `Convolution` which contains plus distributions. For example the comparison with `NConvolution` (see example above) may look as follows:

```
In[9]   :=  Convolution[PlusDistribution[0,1-x],PlusDistribution[0,1-x],x]
```

Out[9] = $2\left[\frac{\log(1-x)}{1-x}\right]_+ - \frac{\mathrm{H}_0(x)}{1-x} - \frac{1}{6}\pi^2\delta(1-x)$

```
In[10]  :=  NEval[%,x,3/10]
```

Out[10] = $2.\left[\frac{\log(1-x)}{1-x}\right]_+ - 1.64493\,\delta(1-x) + 1.71996$

`NEval[`f`, `x`, `a`, `n`]` attempts to output the result with n-digit precision.

`Convolution` utilizes Mellin transforms and inverse Mellin transforms for computing convolutions. `MT` package also provides functions that allow users to find Mellin transforms and inverse Mellin transforms:

- `MTMellinn[`f`, `x`, `n`]` computes the Mellin transform with respect to x of the expression f:

$$M_n\big[f(x)\big] = \int_0^1 dx\, x^{n-1} f(x)\,, \tag{23}$$

which is illustrated in the following example:

```
In[11]  := MTMellinn[HPL[{1},x]/(1+x),x,n]
```

Out[11] = $(-1)^{n-1}\,\mathrm{S}_{-1,1}(n-1)+\frac{1}{12}\pi^2(-1)^{n-1}-\frac{1}{2}(-1)^{n-1}\log^2(2)$

```
In[12]  := MTMellinn[(HPL[{2},x]-HPL[{2},1])/(1-x),x,n]
```

Out[12] = $\mathrm{S}_{2,1}(n-1)-2\zeta(3)$

```
In[13]  := MTMellinn[PlusDistribution[2,1-x],x,n]
```

Out[13] = $-2\,\mathrm{S}_{1,1,1}(n-1)$

- `NMTMellinn[`f`, `x`, `a`]` numerically computes the Mellin transform with respect to x of the expression f with $n = a$.

```
In[14]  := NMTMellinn[HPL[{1},x]/(1+x),x,12]
```

Out[14] = 0.133014

```
In[15]  := NMTMellinn[(HPL[{2},x]-HPL[{2},1])/(1-x),x,12]
```

Out[15] = -0.351258

```
In[16]  := NMTMellinn[PlusDistribution[2,1-x],x,12]
```

Out[16] = -14.684

`NMTMellinn` accepts the following options (see also `NConvolution`): `AccuracyGoal`, `Constants`, `MaxPoints`, `MaxRecursion`, `Method`, `MinRecursion`, `PrecisionGoal`, and `WorkingPrecision`.

- `MTInverse[`f`, `x`, `n`]` gives the inverse Mellin transform with respect to n of the expression f.

```
In[17]  := MTInverse[HSum[{2,1},n]/n,x,n]
```

Out[17] = $\frac{1}{6}\pi^2\,\mathrm{H}_1(x)-\mathrm{H}_{1,2}(x)$

```
In[18]  := MTInverse[(-1)^n*HSum[{-1},n]/n,x,n]
```

Out[18] = $-\log(2)\,\texttt{MTInverse}\left(\frac{(-1)^n}{n},x,n\right)-\mathrm{H}_{-1}(x)+\log(2)$

In the latter example the inverse Mellin transform cannot be expressed in terms of HPLs and is instead reduced to a simpler one.

In the following we list further functions defined in `MT` package which may be useful for users:

- `MTPlusToDReg[`f`]` converts plus distributions in the expression f into regularized derivatives.

```
In[19]    := MTPlusToDReg[PlusDistribution[3,1-x]]
```

Out[19] = $-6\,\hat{\partial}_x\,\mathrm{H}_{1,1,1,1}(x)$

- `MTDRegToPlus[`f`]` converts regularized derivatives in the expression f into plus distributions.

```
In[20]    := MTDRegToPlus[DReg[HPL[{1,1,1,1},x],x]]
```

Out[20] = $-\frac{1}{6}\left[\frac{\log^3(1-x)}{1-x}\right]_+$

- `MTPlusSimplify[`f`]` performs transformations on terms in the expression f containing generalized functions such that their coefficients become constants:

$$\delta(1-x)f(x) = \delta(1-x)f(1), \tag{24}$$

$$\left[\frac{\ln^k(1-x)}{1-x}\right]_+ f(x) = \left[\frac{\ln^k(1-x)}{1-x}\right]_+ f(1) + \frac{\ln^k(1-x)}{1-x}\big[f(x)-f(1)\big]. \tag{25}$$

```
In[21]    := MTPlusSimplify[PlusDistribution[1,1-x]*HPL[{2},x]]
```

Out[21] = $\frac{1}{6}\pi^2\left[\frac{\log(1-x)}{1-x}\right]_+ + \frac{\mathrm{H}_1(x)\big(\pi^2-6\,\mathrm{H}_2(x)\big)}{6(1-x)}$

- `MTHarmonize[`f`, `n`]` shifts the argument of harmonic sums in the expression f such that each term in the result individually depends on a single argument $n+i$.

```
In[22]    := MTHarmonize[HSum[{1,-1},n+2]/n,n]
```

Out[22] = $\frac{\mathrm{S}_{1,-1}(n)}{n} + \frac{3\,\mathrm{S}_{-1}(n)}{2n} - \frac{\mathrm{S}_{-1}(n+1)}{n+1} - \frac{\mathrm{S}_{-1}(n+2)}{2(n+2)} - \frac{5(-1)^n}{4n} - \frac{3(-1)^{n+1}}{2(n+1)} - \frac{(-1)^{n+2}}{4(n+2)}$

- `MTNormalize[`f`, `n`]` normalizes the argument of harmonic sums in the expression f such that they have n as argument.

```
In[23]    := MTNormalize[HSum[{-1,2},n+1],n]
```

Out[23] = $\mathrm{S}_{-1,2}(n) - \frac{(-1)^n\,\mathrm{S}_2(n)}{n+1} - \frac{(-1)^n}{(n+1)^3}$

- `MTProductExpand[`f`]` substitutes products of harmonic sums in the expression f with harmonic sums of higher weights. For a discussion of the corresponding algebra see, e.g., Appendix B of [10].

```
In[24]    := MTProductExpand[HSum[{2},n]*HSum[{-1},n]]
```

Out[24] = $\mathrm{S}_{-1,2}(n) + \mathrm{S}_{2,-1}(n) - \mathrm{S}_{-3}(n)$

- `CancelMTInverse[f]` tries to cancel unevaluated inverse Mellin transforms in the expression f.

```
In[25]  := MTInverse[HSum[{-1},n+1],x,n] - MTInverse[HSum[{-1},n] - (-1)^n/(n+1),x,n]
```

Out[25] = $x\,\texttt{MTInverse}\big(\mathrm{S}_{-1}(n),x,n\big) - \texttt{MTInverse}\big(\mathrm{S}_{-1}(n),x,n\big) - x\,\texttt{MTInverse}\left(\frac{(-1)^n}{n},x,n\right)$

```
In[26]  := CancelMTInverse[%]
```

Out[26] = 0

4. Example 1: collinear singularities for Higgs boson production at the LHC

After the discovery of a new Higgs boson-like particle at the LHC it is now of primary importance to measure its properties like cross sections, branching ratios and couplings with high precision. Current experimental results are compatible with a Standard Model Higgs boson.

The main production mechanism for the latter is the gluon fusion process. Although even second order corrections have been computed for this process (see Refs. [17, 18] and references therein) the perturbative uncertainties are still of the order of 10% and thus it would be desirable to have the next term in the perturbative expansion. `MT` can compute the related subtraction terms for the collinear divergences originating from radiation of partons off initial-state particles.

The N^3LO convolutions to this process have already been discussed in Refs. [4, 5] where results for all convolutions are provided in electronic form. Thus let us at this point only provide an example which demonstrates the use of `MT`.

The results for the partonic cross sections up to NNLO expanded sufficiently deeply in ϵ such that finite results at N^3LO can be obtained are listed in the file `sig_tilde_LO_NLO_NNLO.m` which can be found on the webpage [19] (see also Ref. [4]). As an example we consider the convolution of $\tilde{\sigma}_{qg}^{(1)}/x$ with $P_{gg}^{(1)}$ and $P_{\mathrm{ns}}^{(1)}$ which is computed as[2]

[2] In the examples we present in this and the next Section we do not show the complete output but abbreviate it using ellipses.

```
In[1]   := <<MT`

In[2]   := <<sig_tilde_LO_NLO_NNLO.m

In[3]   := <<splitfnsMVV.m

In[4]   := Convolution[rsigc["NLO", "qg"]/x, Pgg1/.splitfnsMVV, Pqq1/.splitfnsMVV, x];

In[5]   := Collect[%, {ep, HPL[__], PlusDistribution[__], nl}, Together]
```

Out[5] =
$$\ldots + \left(\frac{4(8x^3+11x^2+56x-49)}{9x} - \frac{8\mathrm{nl}(x^2-2x+2)}{27x}\right)\mathrm{H}_{1,0}(x) + \ldots$$
$$+\,\mathrm{ep}\Bigg(\ldots + \mathrm{H}_2(x)\Bigg(-\frac{4(5x^2+2x+4)\,\mathrm{H}_{0,0}(\frac{\mathrm{mu}^2}{\mathrm{Mh}^2})}{3x} - \frac{2\mathrm{nl}(9x^2-26x+7)}{27x}$$
$$+\,\frac{256x^3+237\pi^2x^2+432x^2-78\pi^2x+2832x+276\pi^2-1129}{27x}\Bigg) + \ldots\Bigg)$$
$$+\,\mathrm{ep}^2\Bigg(\ldots - \frac{4(9x^2+6x+4)\,\mathrm{H}_5(x)}{3x} + \ldots + \frac{1}{19440x}(-276480x^3\zeta(3) + 2688\pi^4x^3 + \ldots$$
$$-\,1538205)\Bigg)$$

Note that the number of massless quarks both in `sig_tilde_LO_NLO_NNLO.m` and `splitfnsMVV` is denoted by n_l.

In analogy to this example it is possible to obtain all convolution contributions for the Higgs boson production by combining results from `sig_tilde_LO_NLO_NNLO.m` and `splitfnsMVV.m`, including multiple convolutions.

5. Example 2: collinear singularities for Drell-Yan production

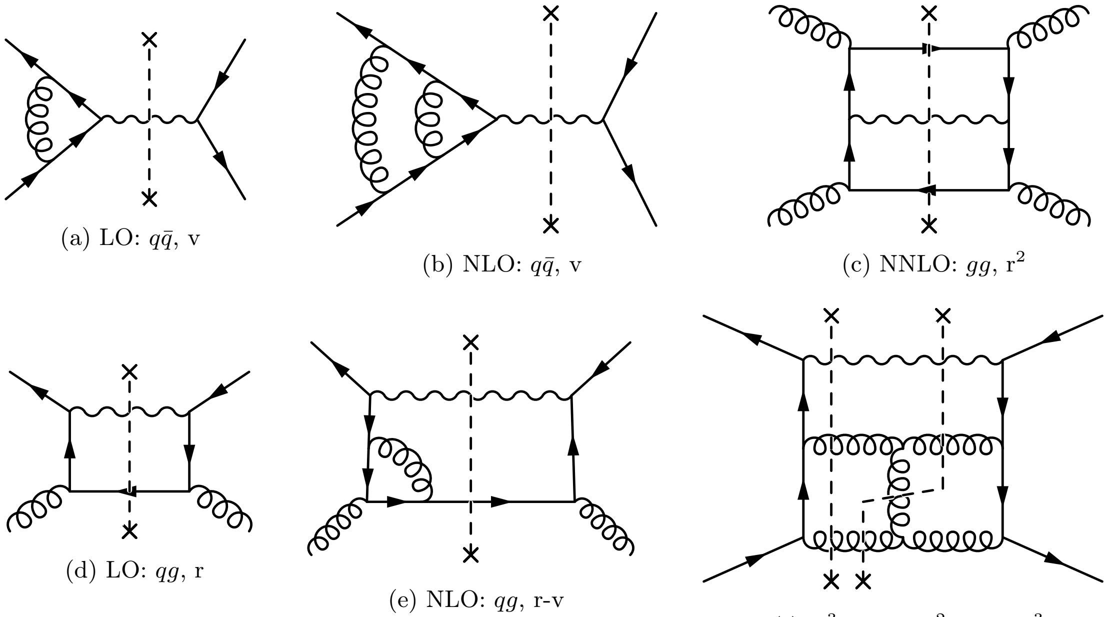
The Drell-Yan process, i.e., the production of a lepton pair in hadronic collisions mediated by a vector boson, constitutes an important benchmark process at hadron colliders. In particular, it provides information about the PDFs and is a useful tool in searches for heavier gauge bosons by examining the invariant mass of the produced leptons.

In the following we briefly discuss the computation of the total cross section up to NNLO with the emphasis on higher-order terms in ϵ. The results are later used in order to evaluate the collinear subtraction terms at N^{3}LO using `MT`. In our discussion we discard all contributions which contain a one-loop triangle diagram with two gluons and a gauge boson as external particles as subgraph since they cancel after summing over all quarks of a family. Thus, we can restrict ourselves to the use of naive anti-commuting γ_5 in the axial-vector coupling of the W or Z boson.

For the calculation of the total cross section we apply the same techniques as for the Higgs production [20, 21], i.e., we use the optical theorem and consider the imaginary part of the forward-scattering amplitude by cutting the gauge boson, light quark and gluon lines. It is convenient to treat the purely virtual corrections separately since in that case only the cut through the gauge boson is considered, see also Fig. 1. The NNLO expression for the resulting quark form factor can be found in Refs. [22, 23, 24, 25], N^{3}LO results are presented in Refs. [26, 27, 28].

Sample Feynman diagrams for the single- and double-real corrections, considered simultaneously, can be found in Fig. 1. At NLO and NNLO the following channels have to be considered

- NLO: $q\bar{q}$ and qg,



(a) LO: $q\bar{q}$, v

(b) NLO: $q\bar{q}$, v

(c) NNLO: gg, r^2

(d) LO: qg, r

(e) NLO: qg, r-v

(f) N^{3}LO: $q\bar{q}$, r^2-v and r^3

Figure 1: Sample Feynman diagrams up to N^{3}LO contributing to the Drell-Yan process. The wiggly lines denote a generic vector boson, i.e., γ, W, or Z. The dashed lines attached to crosses mark the considered cut(s) through a massive vector boson and additional massless particles. The captions state the perturbative order, the channel and the type of contributions ("r" for real, "v" for virtual, or their interference).

- NNLO: $q\bar{q}$, qg, gg, qq, qq',

where q' denotes a quark different from q and the channels involving $\bar{q}$ are not listed separately.

The first step in the evaluation of the Feynman integrals is the reduction to master integrals using the Laporta algorithm [29]. We use our own `C++` implementation for this step. The NNLO master integrals are taken from Ref. [10] (see also [30]) which guarantees that the final result is available including $\mathcal{O}(\epsilon)$ terms.

We present our result using the same notation as in Ref. [31], but set equal the renormalization and the factorization scale, which does not reduce the complexity of the convolutions to be calculated. In particular, we provide terms up to $\mathcal{O}(\epsilon^2)$ for the NLO quantities $\Delta^{(1)}_{q\bar{q}}, \Delta^{(1)}_{qg}$ of Eqs. (B.2), (B.17) and up to $\mathcal{O}(\epsilon)$ for the NNLO results in Eqs. (B.7), (B.18), (B.21), (B.22), (B.24)-(B.26). For $\epsilon = 0$ our results agree with Refs. [31, 32]. We note that, in order to find agreement we had to set the color factor $T_F = 1/2$ in some cases. Moreover in the case of the gluon-gluon contribution the color factor $C_A C_F / T_F$ has been attached to the labels `coeff[]` that will be discussed below. We refrain from explicitly listing analytic results in the paper but provide computer-readable expressions in Ref. [33] (see also Appendix).

It is convenient to identify our partonic cross sections with results of Eq. (A.20) from Ref. [31] which provides a complete expression for the hadronic cross section. The partonic results are easily obtained by removing in that equation the convolutions with the parton density functions, and multiplying by x. Furthermore, in our expressions the summation over initial-state quark and anti-quark flavours is not contained. However, our expressions in the file `sig_DY_LO_NLO_NNLO.m` contain the prefactors of the quantities $x\Delta(x)$ collected in `coeff[]`. In Tab. 2 we provide translation rules from our analytic expressions to the notation used in Eq. (A.20) of Ref. [31].

channel	sig_DY_LO_NLO_NNLO.m	lines of (A.20) in [31]	contains
$\tilde{\sigma}^{(0)}_{q\bar{q}}$	rsigc["LO", "qb"]	2	$\Delta^{(0)}_{q\bar{q}} = \delta(1-x)$
$\tilde{\sigma}^{(1)}_{q\bar{q}}$	rsigc["NLO", "qb"]	2	$\Delta^{(1)}_{q\bar{q}}$
$\tilde{\sigma}^{(1)}_{qg}$	rsigc["NLO", "qg"]	6	$\Delta^{(1)}_{qg}$
$\tilde{\sigma}^{(2)}_{gg}$	rsigc["NNLO", "gg"]	11	$\Delta^{(2)}_{gg}$
$\tilde{\sigma}^{(2)}_{q\bar{q}}$	rsigc["NNLO", "qb"]	2-5	$\Delta^{(2)}_{q\bar{q}}, \Delta^{(2)}_{q\bar{q},B^2}, \Delta^{(2)}_{q\bar{q},BC}, \Delta^{(2),\mathrm{V}}_{q\bar{q},AB}$
$\tilde{\sigma}^{(2)}_{qg}$	rsigc["NNLO", "qg"]	6	$\Delta^{(2)}_{qg}$
$\tilde{\sigma}^{(2)}_{qq'}$	rsigc["NNLO", "qp"]	7,8	$\Delta^{(2)}_{qq,C^2}, \Delta^{(2),\mathrm{V}}_{q\bar{q},CD}$
$\tilde{\sigma}^{(2)}_{qq}$	rsigc["NNLO", "qq"]	9,10	$\Delta^{(2)}_{qq,CE}, \Delta^{(2)}_{qq,CF}$

Table 2: The second column contains the notation for the order and channel shown in the first column as used in the file sig_DY_LO_NLO_NNLO.m. The third column indicates in which lines of Eq. (A.20) from Ref. [31] one can find the corresponding expression. The last column lists the expressions of Ref. [31] one has to consider in order to get our result for the partonic cross section of the second column.

We are now in the position to evaluate all convolutions needed up to N^{3}LO using MT. As a first example we consider the convolution

$$\frac{\tilde{\sigma}^{(1)}_{q\bar{q}}}{x} \otimes P^{(1)}_{\mathrm{ns}}\,, \tag{26}$$

which is a contribution to NNLO. Let us emphasize once more that $\tilde{\sigma}^{(1)}_{q\bar{q}}$ is proportional to $x\Delta^{(1)}_{q\bar{q}}$. The corresponding Mathematica session looks as follows:

```
In[1]   :=  <<MT`

In[2]   :=  <<sig_DY_LO_NLO_NNLO.m

In[3]   :=  <<splitfnsMVV.m

In[4]   :=  Convolution[(rsigc["NLO", "qb"]/x)/.{CF->4/3, CA->3, TF->1/2},
                      Pqq1/.splitfnsMVV, x];

In[5]   :=  Collect[%, {coeff[_], ep, Lv, HPL[__], PlusDistribution[__],nl}, Together]

Out[5]  =
```

$$\mathrm{coeff}(\text{"Cii[qi, bj]} * (\text{vi\^{}2} + \text{ai\^{}2}) * \text{alphas/Pi"})\Bigg(\dots - \frac{32(2x^2+1)\,\mathrm{H}_2(x)}{9(x-1)} + \dots + \mathrm{Lv}\bigg(\dots - \frac{32}{9}(x+1)\,\mathrm{H}_1(x) + \frac{2}{27}(8\pi^2-27)\delta(1-x) + \dots\bigg) + \dots$$
$$+\,\mathrm{ep}\bigg(\dots - \frac{256}{9}(x+1)\,\mathrm{H}_{1,1,1}(x) + \dots + \frac{4}{27}(-96\zeta(3) - 48 + 19\pi^2)\Big[\frac{1}{1-x}\Big]_+ - \frac{2(-72x^2\zeta(3) + \dots + 60)}{27(x-1)}\bigg)$$
$$+\,\mathrm{ep}^2\bigg(\dots - \frac{320(x^2+1)\,\mathrm{H}_{1,2,1}(x)}{9(x-1)} + \dots + \mathrm{Lv}^3\Big(\dots - \frac{16}{27}(x+1)\,\mathrm{H}_1(x) + \frac{1}{81}(8\pi^2-27)\delta(1-x) + \dots\Big) + \dots + \frac{160}{27}\Big[\frac{\log^4(1-x)}{(1-x)}\Big]_+ + \dots\bigg)\Bigg)$$

As one can see, all prefactors are collected in the arguments of the function coeff[]. Furthermore, the final result is available up to order ϵ^2 which is needed since the convolution with $P_{\rm ns}^{(1)}$ is accompanied by a pole in ϵ and the NNLO expression gets multiplied by a renormalization constant which also has an $1/\epsilon$-term.

One of the most involved convolutions which we have to consider is

$$\frac{\tilde{\sigma}_{q\bar{q}}^{(2)}}{x} \otimes P_{\rm ns}^{(1)} \,, \tag{27}$$

since $\sigma_{q\bar{q}}^{(2)}$ contains HPLs up to order four and plus distributions up to $\left[\frac{\ln^4(1-x)}{1-x}\right]_+$. Using MT the calculation looks as follows

```
In[6]  := Convolution[(rsigc["NNLO","qb"]/x)/.{CF->4/3,CA->3,TF->1/2},
                     Pqq1/.splitfnsMVV,x];

In[7]  := Collect[%,{coeff[_],ep,Lv,HPL[__],PlusDistribution[__],nl},Together]
```

Out[7] =

$$\begin{aligned}
&\mathrm{coeff}(\text{"delta[i, j]} * \text{Sum[k, l, Q]} * \text{Cff[qk, bl]} * (\text{vk\^{}2} + \text{ak\^{}2}) * \text{alphas\^{}2/Pi\^{}2"})(\ldots) \\
&+ \mathrm{coeff}(\text{"delta[i, j]} * \text{Sum[k, QB]} * (\text{Cif[qi, bk]} + \text{Cif[bi, qk]}) * (\text{vi\^{}2} + \text{ai\^{}2}) \\
&* \text{alphas\^{}2/Pi\^{}2"})(\ldots) \\
&+ \mathrm{coeff}(\text{"Cii[qi, bj]} * (\text{vi\^{}2} + \text{ai\^{}2}) * \text{alphas\^{}2/Pi\^{}2"})\Bigg(\\
&\ldots + \left(\frac{8\mathrm{nl}(11\mathrm{x}^2+12)}{27(\mathrm{x}-1)} - \frac{4(605\mathrm{x}^2-922\mathrm{x}+617)}{27(\mathrm{x}-1)}\right)\mathrm{H}_{1,2}(\mathrm{x}) \\
&+ \ldots + \mathrm{Lv}^2\left(\ldots - \frac{2}{27}\mathrm{nl}(\mathrm{x}+5) + \ldots\right) + \ldots \\
&+ \mathrm{ep}\Bigg(\ldots - \frac{4(593\mathrm{x}^2+635)\,\mathrm{H}_{1,0,0,0,0}(\mathrm{x})}{27(\mathrm{x}-1)} - \frac{4(1411\mathrm{x}^2+1507)\,\mathrm{H}_{1,1,0,0,0}(\mathrm{x})}{27(\mathrm{x}-1)} + \ldots \\
&+ \left(\frac{464\mathrm{nl}}{243} + \frac{8}{243}(1712\pi^2-429)\right)\left[\frac{\log^3(1-\mathrm{x})}{(1-\mathrm{x})}\right]_+ + \left(\frac{40}{27} - \frac{80\mathrm{nl}}{27}\right)\left[\frac{\log^4(1-\mathrm{x})}{(1-\mathrm{x})}\right]_+ \\
&- \frac{1024}{27}\left[\frac{\log^5(1-\mathrm{x})}{(1-\mathrm{x})}\right]_+ + \ldots\Bigg)\Bigg)
\end{aligned}$$

where the notation for the symbols is given in the Appendix.

6. Summary

The main purpose of this paper is the presentation of the Mathematica package MT which can be used to compute convolution integrals. An algorithm has been implemented based on Mellin transformation and the introduction of generalized derivatives which allows for a simultaneous treatment of HPLs and delta and plus distributions on the same footing. MT contains several functions to perform (inverse) Mellin transforms and manipulations of harmonic sums and plus distributions. Furthermore, Mellin transformations and convolutions can also be performed numerically.

To exemplify the functionality of MT we have considered all convolution integrals to the Higgs boson production and Drell-Yan process up to N^{3}LO in QCD perturbation theory. As a by-product the NNLO Drell-Yan cross section has been computed including $\mathcal{O}(\epsilon)$ terms.

MT can be downloaded from the website [33] which also contains the splitting functions and the partonic cross sections for the Drell-Yan process in computer-readable form. The partonic cross sections for Higgs production can be found on the website [19].

`ep`	ϵ
`nl`	n_l
`CF`	C_F
`CA`	C_A
`TF`	T_F
`Lv`	$\ln(\mu^2/Q^2)$
`coeff[]`	Labels in the argument are described in Tab. 4.

Table 3: Notation used in `sig_DY_LO_NLO_NNLO.m`.

`alphas`	α_s
`Sum[k,QB]`	$\sum_{k\in Q,\bar{Q}}$
`qi`	q_i
`bi`	$\bar{q}_i$
`Cii[qi,bj]`	coupling matrix $C^{\text{ii}}(q_i,\bar{q}_j)$ to vector bosons
`Cif[qi,qj]`	coupling matrix $C^{\text{if}}(q_i,q_j)$ to vector bosons
`Cff[qk,bl]`	coupling matrix $C^{\text{ff}}(q_k,\bar{q}_l)$ to vector bosons
`vi`	v_i
`ai`	a_i
`delta[i,j]`	δ_{ij}

Table 4: Symbolic notation used in the argument of `coeff[]` in `sig_DY_LO_NLO_NNLO.m`. For further explanations see also Appendix A of Ref. [31].

Acknowledgments

This work is supported by the Deutsche Forschungsgemeinschaft in the Sonderforschungsbereich/Transregio 9 "Computational Particle Physics".

Appendix: Description of `sig_DY_LO_NLO_NNLO.m`

The Mathematica file `sig_DY_LO_NLO_NNLO.m` contains the partonic cross sections for Drell-Yan production at LO, NLO and NNLO encoded in the functions listed in Tab. 2. The meaning of the symbols used in `sig_DY_LO_NLO_NNLO.m` is explained in Tabs. 3 and 4 where $C_F = 4/3$, $C_A = 3$ and $T_F = 1/2$ are QCD color factors, n_l counts the number of massless quarks and $\epsilon = (4-D)/2$ is the regularization parameter with D being the space-time dimension. μ is the renormalization scale and $\sqrt{Q^2}$ is the invariant mass of the produced lepton pair.

[1] R. K. Ellis, W. J. Stirling and B. R. Webber, Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol. **8** (1996) 1.

[2] J. Blumlein and S. Kurth, Phys. Rev. D **60** (1999) 014018 [arXiv:hep-ph/9810241].

[3] J. Ablinger, J. Blumlein and C. Schneider, J. Math. Phys. **52** (2011) 102301 [arXiv:1105.6063 [math-ph]].

[4] M. Höschele, J. Hoff, A. Pak, M. Steinhauser and T. Ueda, Phys. Lett. B **721** (2013) 244 [arXiv:1211.6559 [hep-ph]].

[5] S. Buehler and A. Lazopoulos, arXiv:1306.2223 [hep-ph].

[6] G. Altarelli and G. Parisi, Nucl. Phys. B **126** (1977) 298.

[7] G. Curci, W. Furmanski and R. Petronzio, Nucl. Phys. B **175** (1980) 27.

[8] S. Moch, J. A. M. Vermaseren and A. Vogt, Nucl. Phys. B **688** (2004) 101 [hep-ph/0403192].

[9] A. Vogt, S. Moch and J. A. M. Vermaseren, Nucl. Phys. B **691** (2004) 129 [hep-ph/0404111].

[10] A. Pak, M. Rogal and M. Steinhauser, JHEP **1109** (2011) 088 [arXiv:1107.3391 [hep-ph]].

[11] J. A. M. Vermaseren, Int. J. Mod. Phys. A **14** (1999) 2037 [hep-ph/9806280].

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

[13] `http://www.nikhef.nl/~form/maindir/packages/harmpol/`.

[14] D. Maitre, Comput. Phys. Commun. **174** (2006) 222 [hep-ph/0507152].

[15] D. Maitre, Comput. Phys. Commun. **183** (2012) 846 [hep-ph/0703052].

[16] C. Anastasiou, K. Melnikov and F. Petriello, Nucl. Phys. B **724** (2005) 197 [hep-ph/0501130].

[17] S. Dittmaier, C. Mariotti, G. Passarino, R. Tanaka *et al.*, [LHC Higgs Cross Section Working Group Collaboration], arXiv:1101.0593 [hep-ph].

[18] S. Dittmaier, C. Mariotti, G. Passarino, R. Tanaka *et al.*, [LHC Higgs Cross Section Working Group Collaboration], arXiv:1201.3084 [hep-ph].

[19] `http://www.ttp.kit.edu/Progdata/ttp12/ttp12-45`

[20] C. Anastasiou and K. Melnikov, Nucl. Phys. B **646** (2002) 220, arXiv:hep-ph/0207004.

[21] A. Pak, M. Rogal and M. Steinhauser, JHEP **1002** (2010) 025 [arXiv:0911.4662 [hep-ph]].

[22] G. Kramer and B. Lampe, Z. Phys. C **34** (1987) 497 [Erratum-ibid. C **42** (1989) 504].

[23] T. Matsuura and W. L. van Neerven, Z. Phys. C **38** (1988) 623.

[24] T. Matsuura, S. C. van der Marck and W. L. van Neerven, Nucl. Phys. B **319** (1989) 570.

[25] T. Gehrmann, T. Huber and D. Maitre, Phys. Lett. B **622** (2005) 295 [arXiv:hep-ph/0507061].

[26] P. A. Baikov, K. G. Chetyrkin, A. V. Smirnov, V. A. Smirnov and M. Steinhauser, Phys. Rev. Lett. **102** (2009) 212002 [arXiv:0902.3519 [hep-ph]].

[27] T. Gehrmann, E. W. N. Glover, T. Huber, N. Ikizlerli and C. Studerus, JHEP **1006** (2010) 094 [arXiv:1004.3653 [hep-ph]].

[28] R. N. Lee, A. V. Smirnov and V. A. Smirnov, JHEP **1004** (2010) 020 [arXiv:1001.2887 [hep-ph]].

[29] S. Laporta, Int. J. Mod. Phys. A **15** (2000) 5087 [hep-ph/0102033].

[30] C. Anastasiou, S. Buehler, C. Duhr and F. Herzog, JHEP **1211** (2012) 062 [arXiv:1208.3130 [hep-ph]].

[31] R. Hamberg, W. L. van Neerven and T. Matsuura, Nucl. Phys. B **359** (1991) 343 [Erratum-ibid. B **644** (2002) 403].

[32] R. V. Harlander and W. B. Kilgore, Phys. Rev. Lett. **88** (2002) 201801, arXiv:hep-ph/0201206.

[33] `http://www-ttp.kit.edu/Progdata/ttp13/ttp13-27`