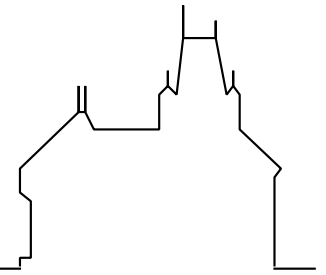


**RISC-Linz**

Research Institute for Symbolic Computation  
Johannes Kepler University  
A-4040 Linz, Austria, Europe



# Extensions of the AZ-algorithm and the Package MultiIntegrate

**J. Ablinger**

January 2021

RISC-Linz Report Series No. 21-02

*Editors: RISC-Linz Faculty*

B. Buchberger, R. Hemmecke, T. Jebelean, T. Kutsia, G. Landsmann, P. Paule,  
V. Pillwein, N. Popov, J. Schicho, C. Schneider, W. Schreiner, W. Windsteiger,  
F. Winkler.

# Extensions of the AZ-algorithm and the Package MultiIntegrate

Jakob Ablinger

**Abstract** We extend the (continuous) multivariate Almkvist-Zeilberger algorithm in order to apply it for instance to special Feynman integrals emerging in renormalizable Quantum field Theories. We will consider multidimensional integrals over hyperexponential integrals and try to find closed form representations in terms of nested sums and products or iterated integrals. In addition, if we fail to compute a closed form solution in full generality, we may succeed in computing the first coefficients of the Laurent series expansions of such integrals in terms of indefinite nested sums and products or iterated integrals. In this article we present the corresponding methods and algorithms. Our Mathematica package `MultiIntegrate`, can be considered as an enhanced implementation of the (continuous) multivariate Almkvist Zeilberger algorithm to compute recurrences or differential equations for hyperexponential integrands and integrals. Together with the summation package `Sigma` and the package `HarmonicSums` our package provides methods to compute closed form representations (or coefficients of the Laurent series expansions) of multidimensional integrals over hyperexponential integrands in terms of nested sums or iterated integrals.

## 1 Introduction

The Almkvist-Zeilberger was first formulated by Apagodu and Zeilberger [14, 32] and has later been refined and generalized [2, 24, 25, 26, 29]. It attracted attention in renormalizable Quantum Field Theory in the frame of the calculation of Feynman integrals.

In the following we briefly want to summarize the structure of those integrals. The very general class of Feynman integrals which are, for instance, considered in [18]

---

Jakob Ablinger  
Johannes Kepler University Linz, Research Institute for Symbolic Computation (RISC), Altenberger  
Straße 69, A-4040, Linz, Austria, e-mail: [Jakob.Ablinger@risc.jku.at](mailto:Jakob.Ablinger@risc.jku.at)

are of relevance for many physical processes at high energy colliders, such as the Large Hadron Collider and others. The considered integrals are two-point Feynman integrals in  $D$ -dimensional Minkowski space with one time- and  $(D - 1)$  Euclidean space dimensions,  $\varepsilon = D - 4$  and  $\varepsilon \in \mathbb{R}$  with  $|\varepsilon| \ll 1$  of the following structure:

$$\mathcal{I}(\varepsilon, N, p) = \int \frac{d^D p_1}{(2\pi)^D} \cdots \int \frac{d^D p_k}{(2\pi)^D} \frac{\mathcal{N}(p_1, \dots, p_k; p; m_1 \dots m_k; \Delta, N)}{(-p_1^2 + m_1^2)^{l_1} \dots (-p_k^2 + m_k^2)^{l_k}} \prod_V \delta_V. \quad (1)$$

They can be shown to obey difference equations with respect to  $N$ , see, e.g., [17]. In (1) the external momentum  $p$  and the loop momenta  $p_i$  denote  $D$ -dimensional vectors,  $m_i > 0, m_i \in \mathbb{R}$  are scalars (masses),  $m_i \in \{0, M\}$ ,  $k, l_i \in \mathbb{N}$ ,  $k \geq 2, l_i \geq 1$ , and  $\Delta$  is a light-like  $D$ -vector,  $\Delta \cdot \Delta = 0$ . The numerator function  $\mathcal{N}$  is a polynomial in the scalar products  $p \cdot p_i$ ,  $p_i \cdot p_k$  and of monomials  $(\Delta \cdot p_{(i)})^{n_i}$ ,  $n_i \in \mathbb{N}, n_i \geq 0$ .  $N \in \mathbb{N}$  denotes the spin of a local operator stemming from the light cone expansion, see, e.g., [27] and references therein, which contributes to the numerator function  $\mathcal{N}$  with a polynomial in  $\Delta \cdot p_i$  of maximal degree  $N$ , cf. [15]. Furthermore it is assumed for simplicity that only one of the loops is formed of massive lines. The  $\delta_V$  occurring in (1) are shortcuts for Dirac delta distributions in  $D$  dimensions  $\delta_V = \delta^{(D)}\left(\sum_{l=1}^k a_{V,l} p_l\right), a_{V,l} \in \mathbb{Q}$ .

These integrals are mathematically well defined and in [18] it is shown how they can be mapped onto integrals on the  $m$ -dimensional unit cube with the following structure:

$$\mathcal{I}(\varepsilon, N) = C(\varepsilon, N, M) \int_0^1 dy_1 \dots \int_0^1 dy_m \frac{\sum_{i=1}^k \prod_{l=1}^{r_i} [P_{i,l}(y)]^{\alpha_{i,l}(\varepsilon, N)}}{[Q(y)]^{\beta(\varepsilon)}}, \quad (2)$$

with  $k \in \mathbb{N}$ ,  $r_1, \dots, r_k \in \mathbb{N}$  and where  $\beta(\varepsilon)$  is given by a rational function in  $\varepsilon$ , i.e.,  $\beta(\varepsilon) \in \mathbb{R}(\varepsilon)$ , and similarly  $\alpha_{i,l}(\varepsilon, N) = n_{i,l}N + \bar{\alpha}_{i,l}$  for some  $n_{i,l} \in \{0, 1\}$  and  $\bar{\alpha}_{i,l} \in \mathbb{R}(\varepsilon)$ , see also [21] in the case no local operator insertions are present.  $C(\varepsilon, N, M)$  is a factor, which depends on the dimensional parameter  $\varepsilon$ , the integer parameter  $N$  and the mass  $M$ .  $P_i(y), Q(y)$  are polynomials in the remaining Feynman parameters  $y = (y_1, \dots, y_m)$  written in multi-index notation. In (2) all terms which stem from local operator insertions were geometrically resummed; see [15]. In [18] it was already mentioned that after splitting the integral (2), the integrands fit into the input class of the multivariate Almkvist-Zeilberger algorithm. Hence, if the split integrals are properly defined, they obey homogeneous recurrence relations in  $N$  due to the theorems in [14]. In [18] the integrals of (2) are transformed further to a multi-sum representation, while in this article (and also in [2, 6, 8]) we want to tackle them directly by looking on integrals of the form

$$\mathcal{I}(n) = \mathcal{I}(\varepsilon, n) = \int_{u_d}^{o_d} \cdots \int_{u_1}^{o_1} F(n; x_1, \dots, x_d; \varepsilon) dx_1 \cdots dx_d, \quad (3)$$

with  $d, n \in \mathbb{N}$ ,  $F(n; x_1, \dots, x_d; \varepsilon)$  a hyperexponential term,  $\varepsilon > 0$  a real parameter and  $u_i, o_i \in \mathbb{R} \cup \{-\infty, \infty\}$ .

In [2] we only considered a discrete variable  $n$  but here we will also consider continuous variables, i.e., we will also deal with integrals of the form

$$I(x) = I(\varepsilon, x) = \int_{u_d}^{o_d} \cdots \int_{u_1}^{o_1} F(x; x_1, \dots, x_d; \varepsilon) dx_1 \cdots dx_d, \quad (4)$$

with  $d \in \mathbb{N}$ ,  $x \in \mathbb{R}$ ,  $F(x; x_1, \dots, x_d; \varepsilon)$  a hyperexponential term,  $\varepsilon > 0$  a real parameter and  $u_i, o_i \in \mathbb{R} \cup \{-\infty, \infty\}$ . We will use our package `MultiIntegrate`<sup>1</sup>[2, 8] that can be considered as an enhanced implementation of the multivariate Almkvist Zeilberger algorithm to compute recurrences/differential equations for the integrands and integrals. For solving recurrences `MultiIntegrate` relies on the solver implemented in the packages `Sigma`<sup>1</sup> [38, 39, 40] and `EvaluateMultiSums`<sup>1</sup> [39, 40], while for solving differential equation it relies on the solver implemented in the package `HarmonicSums`<sup>1</sup> [1, 2, 4, 5, 7].

Throughout this article  $\mathbb{K}$  denotes a field with  $\mathbb{Q} \subseteq \mathbb{K}$  (e.g.,  $\mathbb{K} = \mathbb{Q}(\varepsilon)$  forms a rational function field) in which the usual operations can be computed.

The remainder of this article is structured in two main sections. In Section 2 we will recall the multi-variate Almkvist-Zeilberger algorithm and its modifications as presented in [2] to solve integrals of the form (3), while in Section 3 we will present a method based on the continuous Almkvist-Zeilberger algorithm to solve integrals of the form (4). The reason for Section 2 is its similarity to the continuous case in Section 3, in addition it allows us to present a complete picture of the functionality of the package `MultiIntegrate`. However, in order to keep Section 2 short, we omit examples in this section and refer to [2, 6, 8] for further illustrations.

## 2 A fine-tuned multi-variate Almkvist-Zeilberger algorithm

In this section we will recall a method (presented in [2] and [8]) to compute integrals of the form (3), that is based on slight modifications of the multi-variate Almkvist-Zeilberger algorithm [14] and implemented in the package `MultiIntegrate`. The method relies on finding and solving recurrences.

In general, consider the integrand

$$F(n; x_1, \dots, x_d) = P(n; x_1, \dots, x_d) \cdot H(n; x_1, \dots, x_d), \quad (5)$$

with a multivariate polynomial  $P(n; x_1, \dots, x_d) \in \mathbb{K}[n, x_1, \dots, x_d]$  and

$$H(n; x_1, \dots, x_d) = e^{\frac{a(x_1, \dots, x_d)}{b(x_1, \dots, x_d)}} \cdot \left( \prod_{p=1}^P S_p(x_1, \dots, x_d)^{\alpha_p} \right) \cdot \left( \frac{s(x_1, \dots, x_d)}{t(x_1, \dots, x_d)} \right)^n,$$

---

<sup>1</sup> The Mathematica packages `MultiIntegrate`, `HarmonicSums`, `Sigma` and `EvaluateMultiSums` can be downloaded at <https://risc.jku.at/software>

where  $a(x_1, \dots, x_d)$  and  $b(x_1, \dots, x_d) \neq 0$ ,  $s(x_1, \dots, x_d)$  and  $t(x_1, \dots, x_d) \neq 0$  and  $S_p(x_1, \dots, x_d) \in \mathbb{K}[x_1, \dots, x_d]$ , and  $\alpha_p \in \mathbb{K}$ . Such integrands have the property that the logarithmic derivatives are rational, *i.e.*,

$$\frac{D_{x_i} \bar{H}(n; x_1, \dots, x_d)}{\bar{H}(n; x_1, \dots, x_d)} = \frac{q_i(x_1, \dots, x_d)}{r_i(x_1, \dots, x_d)}$$

for some  $q_i(n, x_1, \dots, x_d), r_i(n, x_1, \dots, x_d) \in \mathbb{K}[n, x_1, \dots, x_d]$  and are called *hyper-exponential* in  $x_i$ . Note that this class of integrands covers a big class of Feynman integrals (by choosing the rational function field  $\mathbb{K} = \mathbb{Q}(\varepsilon)$ ) that contains at most one mass [18, 43].

Then due to [14] there exists a non-negative integer  $L$ , there exist  $e_0(n), \dots, e_L(n) \in \mathbb{K}[n]$  (or equivalently from  $\mathbb{K}(n)$ ), *not all zero*, and there also exist  $R_i(n; x_1, \dots, x_d) \in \mathbb{K}(n, x_1, \dots, x_d)$  such that

$$G_i(n; x_1, \dots, x_d) := R_i(n; x_1, \dots, x_d) F(n; x_1, \dots, x_d) \quad (6)$$

satisfy the integrand recurrence

$$\sum_{i=0}^L e_i(n) F(n+i; x_1, \dots, x_d) = \sum_{i=1}^d D_{x_i} G_i(n; x_1, \dots, x_d), \quad (7)$$

where  $D_{x_i}$  stands for the derivative w.r.t  $x_i$ .

## 2.1 The general method

The proof of the existence, and in particular a method to compute such an integrand recurrence (7), is based on the following observation [14]. Fix a non-negative integer  $L$  (with the role given above) and define

$$\bar{H}(n; x_1, \dots, x_d) := e^{\frac{a(x_1, \dots, x_d)}{b(x_1, \dots, x_d)}} \cdot \left( \prod_{p=1}^P S_p(x_1, \dots, x_d)^{\alpha_p} \right) \cdot \frac{s(x_1, \dots, x_d)^n}{t(x_1, \dots, x_d)^{n+L}},$$

Then we have

$$\sum_{i=0}^L e_i(n) F(n+i; x_1, \dots, x_d) = h(x_1, \dots, x_d) \bar{H}(n; x_1, \dots, x_d),$$

where  $h(x_1, \dots, x_d)$  is a polynomial *i.e.*,

$$h(x_1, \dots, x_d) := \sum_{i=1}^L e_i(n) P(n+i, x_1, \dots, x_d) \frac{s(x_1, \dots, x_d)^i}{t(x_1, \dots, x_d)^{i-L}}.$$

and, by construction, the logarithmic derivatives of  $\overline{H}(n; x_1, \dots, x_d)$  are a rational functions in the  $x_i$ , i.e., we have that

$$\frac{D_{x_i} \overline{H}(n; x_1, \dots, x_d)}{\overline{H}(n; x_1, \dots, x_d)} = \frac{q_i(x_1, \dots, x_d)}{r_i(x_1, \dots, x_d)}$$

for explicitly given  $q_i(n, x_1, \dots, x_d), r_i(n, x_1, \dots, x_d) \in \mathbb{K}[n, x_1, \dots, x_d]$ . For  $i = 1, \dots, d$  we make the general ansatz

$$G_i(n; x_1, \dots, x_d) = \overline{H}(n; x_1, \dots, x_d) \cdot r_i(n, x_1, \dots, x_d) \cdot X_i(n; x_1, \dots, x_d). \quad (8)$$

Then it turns out that for  $L$  chosen sufficiently large<sup>2</sup> there exist polynomials  $X_i(n; x_1, \dots, x_d) \in \mathbb{K}[n][x_1, \dots, x_d]$  with  $1 \leq i \leq L$  and polynomials  $e_i(n) \in \mathbb{K}[n]$  (not all zero) such that (7) holds. Motivated by this fact, one searches for these unknowns  $X_i$  and  $e_i$  as follows. Note that the ansatz (7) is equivalent to (see [14])

$$\begin{aligned} & \sum_{i=1}^d [D_{x_i} r_i(x_1, \dots, x_d) + q_i(x_1, \dots, x_d)] \cdot X_i(n; x_1, \dots, x_d) \\ & + r_i(x_1, \dots, x_d) \cdot D_{x_i} X_i(n; x_1, \dots, x_d) \\ & = \sum_{i=0}^L e_i(n) P(n; x_1, \dots, x_d) s(x_1, \dots, x_d)^i t(x_1, \dots, x_d)^{L-i}. \end{aligned} \quad (9)$$

We choose appropriate degree bounds w.r.t. the  $x_1, \dots, x_d$  for the  $X_i$  ( $1 \leq i \leq d$ ) and plug the polynomials with unknown coefficients from  $\mathbb{K}[n]$  (from  $\mathbb{K}(n)$ ) into (9). By coefficient comparison this yields a linear system in  $\mathbb{K}(n)$  with the unknowns  $e_i(n)$  and the unknown coefficients of the polynomials  $X_i$ . Finally, we can seek for a non-trivial solution for (9) and thus for (7). To optimize the search for a non-trivial solution we make use of homomorphic image computations in our implementation. More precisely, we plug in some concrete integers for the parameters and reduce all integer coefficients modulo a prime. If there is no solution in the homomorphic setting, there is no solution in the general setting. By choosing these values sufficiently generically we can also minimize the risk of obtaining a homomorphic solution that does not extend to a general solution. In the end, we clear denominators in  $n$  such that the  $e_i(n)$  turn to polynomials.

If  $F(n; \dots, x_{i-1}, u_i, x_{i+1}, \dots) = 0$  and  $F(n; \dots, x_{i-1}, o_i, x_{i+1}, \dots) = 0$  then

$$\mathcal{I}(n) := \int_{u_d}^{o_d} \cdots \int_{u_1}^{o_1} F(n; x_1, \dots, x_d) dx_1 \cdots dx_d,$$

satisfies the homogeneous linear recurrence equation with polynomial coefficients

---

<sup>2</sup> There exist upper bounds for a particular input. But usually, these bounds are too high and one tries smaller values.

$$\sum_{i=0}^L e_i(n) \mathcal{I}(n+i) = 0. \quad (10)$$

The general method now is straightforward: Given an integrand of the form (5), we can set  $L = 0$ , look for degree bounds for  $X_i(x_1, \dots, x_d)$  and try to find a solution of (10) by coefficient comparison. If we do not find a solution of (10) with not all  $e_i(n)$ 's equal to zero (we stop the calculation if the homomorphic image check fails), we increase  $L$  by one, look for new degree bounds for  $X_i(x_1, \dots, x_d)$  and try again to find a solution of (10). Again, if we do not find a solution with not all  $e_i(n)$ 's equal to zero, we increase  $L$  by one and repeat the process.

The discrete multiple Almkvist-Zeilberger algorithm is implemented in the command `mAZ` of `MultiIntegrate`.

Once we found a recurrence we exploit algorithms from [12, 33, 35, 37] which can constructively decide if a solution with certain initial values is expressible in terms of indefinite nested products and sums. This covers harmonic sums [16, 41], S-sums [4, 31], cyclotomic sums [1] and binomial sums [7, 28] as special cases. In our implementation we make use of the algorithms implemented in the summation package `Sigma`. For details on which solutions can be found using `Sigma`, we refer to [18].

## 2.2 Dealing with non-standard boundary conditions

Unfortunately, in many cases the integrand (5) does not vanish at the integration bounds and we end up in a linear recurrence with a non-trivial inhomogeneous part which can be written as a linear combination of integrals with at least one integral operator less. In the following we will deal with non-standard boundary conditions in two different ways, see [2].

### 2.2.1 Dealing with inhomogeneous recurrences

In [2] a method that deals with the inhomogeneous recurrence similar to [18] can be found. It gives rise to a recursive method. To be more precise, we consider the integral

$$\mathcal{I}(n) := \int_{u_d}^{o_d} \cdots \int_{u_1}^{o_1} F(n; x_1, \dots, x_d) dx_1 \cdots dx_d.$$

Suppose that we found

$$\sum_{i=0}^L e_i(n)F(n+i; x_1, \dots, x_d) = \sum_{i=1}^d D_{x_i} G_i(n; x_1, \dots, x_d) \quad (11)$$

where at least one  $G_i(n; x_1, \dots, x_d)$  does not vanish at the integration limits. By integration with respect to  $x_1, \dots, x_d$  we can deduce that  $\mathcal{I}(n)$  satisfies the inhomogeneous linear recurrence equation

$$\begin{aligned} \sum_{i=0}^L e_i(n)\mathcal{I}(n+i) = & \\ & \sum_{i=1}^d \int_{u_d}^{o_d} \cdots \int_{u_{i-1}}^{o_{i-1}} \int_{u_{i+1}}^{o_{i+1}} \cdots \int_{u_1}^{o_1} O_i(n) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_d \\ & - \sum_{i=1}^d \int_{u_d}^{o_d} \cdots \int_{u_{i-1}}^{o_{i-1}} \int_{u_{i+1}}^{o_{i+1}} \cdots \int_{u_1}^{o_1} U_i(n) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_d \end{aligned}$$

with

$$\begin{aligned} U_i(n) &:= G_i(n; x_1, \dots, x_{i-1}, o_i, x_{i+1}, \dots, x_d) \\ O_i(n) &:= G_i(n; x_1, \dots, x_{i-1}, u_i, x_{i+1}, \dots, x_d). \end{aligned}$$

Note that the inhomogeneous part of the above recurrence equation is a sum of  $2 \cdot d$  integrals of dimension  $d - 1$ , which fit again into the input class of the multiple Almkvist-Zeilberger algorithm. Hence we can apply the algorithms to the  $2 \cdot d$  integrals recursively until we arrive at the base case of one-dimensional integrals for which we have to solve an inhomogeneous linear recurrence relation where the inhomogeneous part is free of integrals. Given the solutions for the one-dimensional integrals we can step by step find the solutions of higher dimensional integrals until we finally find the solution for  $\mathcal{I}(n)$  by solving again an inhomogeneous linear recurrence equation and combining it with the initial values. Note that we have to calculate initial values with respect to  $n$  for all the integrals arising in this process.

Summarizing, with these algorithms we use the following strategy (note that we assume that we are able to compute the initial values for the arising integrals); compare [2, 18]:

### Divide and conquer strategy

1. BASE CASE: If  $\mathcal{I}(n)$  has no integration quantifiers, return  $\mathcal{I}(n)$ .
2. DIVIDE: As worked out above, compute a recurrence relation

$$a_0(n)\mathcal{I}(n) + \cdots + a_d(n)\mathcal{I}(n+d) = h(n) \quad (12)$$

with polynomial coefficients  $a_i(n) \in \mathbb{K}[n]$ ,  $a_m(n) \neq 0$  and the right side  $h(n)$  containing a linear combination of hyperexponential multi-integrals each with less than  $d$  integration quantifiers.



3. CONQUER: Apply the strategy recursively to the simpler integrals in  $h(n)$ . This results in an indefinite nested product-sum expressions  $\tilde{h}(n)$  with

$$\tilde{h}(n) = h(n), \quad \forall n \geq \delta \text{ for some } \delta \in \mathbb{N}. \quad (13)$$

If the method fails to find the  $\tilde{h}(n)$  in terms of indefinite nested product-sum expressions, STOP.

4. COMBINE: Given (12) with (13), compute, if possible,  $\tilde{\mathcal{I}}(n)$  in terms of nested product-sum expressions such that

$$\tilde{\mathcal{I}}(n) = \mathcal{I}(n), \quad \forall n \geq \delta \text{ for some } \delta \in \mathbb{N}. \quad (14)$$

by solving the recurrence.

This divide and conquer strategy is implemented in the command `mAZIntegrate` of `MultiIntegrate`.

*Remark 1* We remark that this approach works nicely, if the initial values of the integrals in the inhomogeneous part can be calculated efficiently. Further details on this approach are given in [2, 3]. We remark further that similar approaches have been explored in [18, 20] and [3] based on [42, 44] and [36], respectively, in order to derive recurrences for hypergeometric multi-sums.

## 2.2.2 Adapting the ansatz to find homogeneous recurrences

In order to avoid the difficulties of inhomogeneous recurrences we adapt the ansatz. Namely, we can always obtain a homogeneous recurrence of the form (10) by changing (8) to

$$G_i(n; x_1, \dots, x_d) = \overline{H}(n; x_1, \dots, x_d) \cdot r_i(x_1, \dots, x_d) \cdot X_i(x_1, \dots, x_d)(x_i - u_i)(x_i - o_i), \quad (15)$$

i.e., the  $G_i$  are forced to vanish at the integration bounds. Then with this Ansatz (9) the underlying linear system turns into

$$\begin{aligned} & \sum_{i=1}^d [D_{x_i} r_i(x_1, \dots, x_d) + q_i(x_1, \dots, x_d)] \cdot X_i(x_1, \dots, x_d)(x_i - u_i)(x_i - o_i) \\ & + r_i(x_1, \dots, x_d) \cdot D_{x_i} X_i(x_1, \dots, x_d)(x_i - u_i)(x_i - o_i) \\ & = \sum_{i=0}^L e_i(N) P(N; x_1, \dots, x_d) s(x_1, \dots, x_d)^i t(x_1, \dots, x_d)^{L-i}. \end{aligned} \quad (16)$$

The general method now is straightforward: Given an integrand of the form (5), we can set  $L = 0$ , look for degree bounds for  $X_i(x_1, \dots, x_d)$  and try to find a solution of (16) by coefficient comparison. If we do not find a solution of (16) with not all  $e_i(n)$ 's equal to zero, we increase  $L$  by one, look for new degree bounds for  $X_i(x_1, \dots, x_d)$  and try again to find a solution of (16). Again, if we do not find a solution with not all  $e_i(n)$ 's equal to zero, we increase  $L$  by one and repeat the process.

Once we found a recurrence we can use the recurrence solver implemented in the summation package `Sigma` to try to solve it.

This strategy is implemented in the command `mAZDirectIntegrate` of `MultiIntegrate`.

*Remark 2* The advantage of this approach is, that we do not have to deal with integrals (and initial conditions) recursively, since the recurrence is homogenous, however the additional conditions on the ansatz might increase the order of the recurrence drastically. In particular, the routine is more robust: no abortion can occur due to problematic integral arising from the recursion.

### 2.3 Computing series expansions of the integrals

Due to time and memory limitations, not finding all solutions of the recurrences or due to missing initial values (in full generality) we might fail to process certain integrals using the methods described in the previous subsection. Therefore, inspired by [18], a method which computes  $\varepsilon$ -expansions of integrals of the form (3) was developed in [2]. In the following we recall this method. We assume that the integral  $I(\varepsilon, n)$  from (3) has a Laurent expansion in  $\varepsilon$  for each  $n \in \mathbb{N}$  with  $n \geq \lambda$  for some  $\lambda \in \mathbb{N}$  and thus it is an analytic function in  $\varepsilon$  throughout an annular region centered by 0 where the pole at  $\varepsilon = 0$  has some order  $K \in \mathbb{Z}$ . Hence we can write it in the form

$$I(\varepsilon, n) = \sum_{k=-K}^{\infty} \varepsilon^k I_k(n). \quad (17)$$

In the following we try to find the first coefficients  $I_t(n), I_{t+1}(n), \dots, I_u(n)$  in terms of indefinite nested product-sum expressions of the expansion

$$I(\varepsilon, n) = I_t(n)\varepsilon^t + I_{t+1}(n)\varepsilon^{t+1} + I_{t+2}(n)\varepsilon^{t+2} + \dots \quad (18)$$

with  $t = -K \in \mathbb{Z}$ . We start by computing a recurrence for  $I(\varepsilon, n)$  in the form

$$\begin{aligned} a_0(\varepsilon, n)J(\varepsilon, n) + a_1(\varepsilon, n)J(\varepsilon, n+1) + \dots + a_d(\varepsilon, n)J(\varepsilon, n+d) \\ = h_{-K}(n)\varepsilon^{-K} + h_{-K+1}(n)\varepsilon^{-K+1} + \dots + h_u(n)\varepsilon^u + \dots \end{aligned} \quad (19)$$

In order to accomplish this task, we can use the methods presented in the previous section. Given the recurrence we exploit an algorithm from [18] which can constructively decide if a formal Laurent series solution with certain initial values is expressible (up to a certain order) in terms of indefinite nested products and sums. This algorithm is implemented in the package `Sigma` and can be summarized as follows (see [18] and compare [2, 8]).

Suppose we are given the linear recurrence (19) of order  $d$  where the  $a_i(\varepsilon, n)$  are polynomials in  $n$  and  $\varepsilon$  and where the inhomogeneous part can be expanded in  $\varepsilon$  up to order  $u$ . Consider a function which has a Laurent series expansion

$$\mathcal{I}(\varepsilon, n) = F_t(n)\varepsilon^t + F_{t+1}(n)\varepsilon^{t+1} + \dots \quad (20)$$

and which is a solution of the given recurrence for all  $n \geq n_0$  for some  $n_0 \in \mathbb{N}$ . Then together with the  $d$  initial values  $F_j(n_0), \dots, F_j(n_0 + d - 1)$  with  $t \leq j \leq u$ , all values  $F_t(n), \dots, F_u(n)$  with  $n \geq n_0$  can be computed provided that the values  $h_i(n)$  for all  $i$  with  $t \leq i \leq u$  and all integers  $n$  with  $n \geq n_0$  can be computed. In addition, if the  $h_t(n), \dots, h_u(n)$  are given explicitly in terms of indefinite nested product-sum expressions, there is an algorithm which decides constructively if the  $F_t(n), \dots, F_u(n)$  can be given in terms of indefinite nested product-sum expressions.

Having such a Laurent series recurrence solver in hand we can combine it with the methods from the previous sections. Let  $\mathcal{I}(\varepsilon, n)$  be a multi-integral of the form (3) and assume that  $\mathcal{I}(\varepsilon, n)$  has a series expansion (18) for all  $n \geq \lambda$  for some  $\lambda \in \mathbb{N}$ . If we succeed in finding a homogeneous differential equation, for instance by using the method from Section 2.2.2 we can directly apply the Laurent series recurrence solver, supposing that we can handle the initial values. This has been exploited in the frame of [8].

This strategy is implemented in the command `mAZExpandedDirectIntegrate` of `MultiIntegrate`.

Of course we can again think of a recursive method to compute the first coefficients (compare [2, 18]), say  $F_t(n), \dots, F_u(n)$  of (18). Note that we have the same advantages and disadvantages as mentioned for the recursive method in Section 2.2.1, but if we assume that we can handle the initial values we can use the following strategy.

#### Divide and conquer strategy

1. BASE CASE: If  $\mathcal{I}(\varepsilon, n)$  has no integration quantifiers, compute the expansion by standard methods.
2. DIVIDE: As worked out before, compute a recurrence relation

$$a_0(\varepsilon, n)\mathcal{I}(\varepsilon, n) + \dots + a_d(\varepsilon, n)\mathcal{I}(\varepsilon, n + d) = h(\varepsilon, n) \quad (21)$$

with polynomial coefficients  $a_i(\varepsilon, n) \in \mathbb{K}[\varepsilon, n]$ ,  $a_m(\varepsilon, n) \neq 0$  and the right side  $h(\varepsilon, n)$  containing a linear combination of hyperexponential multi-integrals each with less than  $d$  integration quantifiers.

3. CONQUER: Apply the strategy recursively to the simpler integrals in  $h(\varepsilon, n)$ . This results in an expansion of the form

$$h(\varepsilon, n) = h_t(n) + h_1(n)\varepsilon + \cdots + h_u(n)\varepsilon^u + O(\varepsilon^{u+1}); \quad (22)$$

if the method fails to find the  $h_t(n), \dots, h_u(n)$  in terms of indefinite nested product-sum expressions, STOP.

4. COMBINE: Given (21) with (22), compute, if possible, the  $F_t(n), \dots, F_u(n)$  of (18) in terms of nested product-sum expressions by using `Sigma`.

This divide and conquer strategy is implemented in the command `mAZ-ExpandedIntegrate` of `MultiIntegrate`.

### 3 A fine-tuned continuous multi-variate Almkvist-Zeilberger algorithm

In this section we present a method to compute integrals of the form (4), that is based on slight modifications of the continuous multi-variate Almkvist-Zeilberger algorithm [14] and implemented in the package `MultiIntegrate`. Unlike in the discrete case, this method relies on finding and solving differential equations. In general, consider the hyperexponential integrand

$$F(x; x_1, \dots, x_d) = P(x; x_1, \dots, x_d) \cdot H(x; x_1, \dots, x_d), \quad (23)$$

with a multivariate polynomial  $P(x; x_1, \dots, x_d) \in \mathbb{K}[x, x_1, \dots, x_d]$  and

$$H(x; x_1, \dots, x_d) = e^{\frac{a(x, x_1, \dots, x_d)}{b(x, x_1, \dots, x_d)}} \cdot \left( \prod_{p=1}^P S_p(x, x_1, \dots, x_d)^{\alpha_p} \right),$$

where  $a(x, x_1, \dots, x_d), b(x, x_1, \dots, x_d)$  and  $S_p(x, x_1, \dots, x_d) \in \mathbb{K}[x, x_1, \dots, x_d]$ , with  $b(x, x_1, \dots, x_d) \neq 0$ , and  $\alpha_p \in \mathbb{K}$ . Then due to [14] there exists a non-negative integer  $L$ , there exist  $e_0(x), e_1(x), \dots, e_L(x) \in \mathbb{K}[x]$  (or equivalently from  $\mathbb{K}(x)$ ), *not all zero*, and there also exist  $R_i(x; x_1, \dots, x_d) \in \mathbb{K}(x, x_1, \dots, x_d)$  such that

$$G_i(x; x_1, \dots, x_d) := R_i(x; x_1, \dots, x_d)F(x; x_1, \dots, x_d) \quad (24)$$

satisfy the integrand differential equation

$$\sum_{i=0}^L e_i(x) D_x^i F(x; x_1, \dots, x_d) = \sum_{i=1}^d D_{x_i} G_i(x; x_1, \dots, x_d). \quad (25)$$

### 3.1 The general method

The proof of the existence, and in particular a method to compute such a differential equation (25), is based on the following observation [14]. Fix a non-negative integer  $L$  (with the role given above), define

$$\overline{H}(x; x_1, \dots, x_d) := \frac{e^{\frac{a(x, x_1, \dots, x_d)}{b(x, x_1, \dots, x_d)}}}{b(x, x_1, \dots, x_d)^{2L}} \cdot \left( \prod_{p=1}^P S_p(x, x_1, \dots, x_d)^{\alpha_p} \right),$$

Then we have

$$\sum_{i=0}^L e_i(x) D_x^i F(x; x_1, \dots, x_d) = h(x, x_1, \dots, x_d) \overline{H}(x; x_1, \dots, x_d).$$

for some polynomial  $h(x, x_1, \dots, x_d)$  that can be determined and, by construction, the logarithmic derivatives of  $\overline{H}(x; x_1, \dots, x_d)$  are rational functions in the  $x_i$ , i.e., we have that

$$\frac{D_{x_i} \overline{H}(x; x_1, \dots, x_d)}{\overline{H}(x; x_1, \dots, x_d)} = \frac{q_i(x, x_1, \dots, x_d)}{r_i(x, x_1, \dots, x_d)}$$

for explicitly given  $q_i(x, x_1, \dots, x_d), r_i(x, x_1, \dots, x_d) \in \mathbb{K}[x, x_1, \dots, x_d]$ .

For  $i = 1, \dots, d$  we make the general ansatz

$$G_i(x; x_1, \dots, x_d) = \overline{H}(x; x_1, \dots, x_d) \cdot r_i(x, x_1, \dots, x_d) \cdot X_i(x; x_1, \dots, x_d). \quad (26)$$

Then it turns out that for  $L$  chosen sufficiently large<sup>3</sup> there exist polynomials  $X_i(x; x_1, \dots, x_d) \in \mathbb{K}[x][x_1, \dots, x_d]$  with  $1 \leq i \leq L$  and polynomials  $e_i(x) \in \mathbb{K}[x]$  (not all zero) such that (25) holds. Motivated by this fact, one searches for these unknowns  $X_i$  and  $e_i$  as follows. Note that the ansatz (25) is equivalent to (see [14])

$$\begin{aligned} & \sum_{i=1}^d [D_{x_i} r_i(x, x_1, \dots, x_d) + q_i(x, x_1, \dots, x_d)] \cdot X_i(x; x_1, \dots, x_d) \\ & + r_i(x, x_1, \dots, x_d) \cdot D_{x_i} X_i(x, x_1, \dots, x_d) = h(x, x_1, \dots, x_d). \end{aligned} \quad (27)$$

Finally, we choose appropriate degree bounds w.r.t. the  $x_1, \dots, x_d$  for the  $X_i$  ( $1 \leq i \leq d$ ) and plug the polynomials with unknown coefficients from  $\mathbb{K}[x]$  (from  $\mathbb{K}(x)$ ) into (27). By coefficient comparison this yields a linear system in  $\mathbb{K}(x)$  with the

<sup>3</sup> There exist upper bounds for a particular input. But usually, these bounds are too high and one tries smaller values.

unknowns  $e_i(x)$  and the unknown coefficients of the polynomials  $X_i$ . Finally, we can seek a non-trivial solution for (27) and thus for (25). In the end, we clear denominators in  $x$  such that the  $e_i(x)$  turn to polynomials.

If  $F(x; \dots, x_{i-1}, u_i, x_{i+1}, \dots) = 0$  and  $F(x; \dots, x_{i-1}, o_i, x_{i+1}, \dots) = 0$  then

$$\mathcal{I}(x) := \int_{u_d}^{o_d} \cdots \int_{u_1}^{o_1} F(x; x_1, \dots, x_d) dx_1 \dots dx_d,$$

satisfies the homogeneous linear differential equation with polynomial coefficients

$$\sum_{i=0}^L e_i(x) D_x^i \mathcal{I}(x) = 0. \quad (28)$$

The general method now is straightforward: Given an integrand of the form (23), we can set  $L = 0$ , look for degree bounds for  $X_i(x, x_1, \dots, x_d)$  and try to find a solution of (28) by coefficient comparison. If we do not find a solution of (28) with not all  $e_i(x)$ 's equal to zero (with homomorphic image testing to decide non-existence efficiently), we increase  $L$  by one, look for new degree bounds for  $X_i(x, x_1, \dots, x_d)$  and try again to find a solution of (28). Again, if we do not find a solution with not all  $e_i(x)$ 's equal to zero, we increase  $L$  by one and repeat the process.

The continuous Almkvist-Zeilberger algorithm is implemented in the command `cmAZ` of `MultiIntegrate`.

Once we found a differential equation we can make use of the differential equation solver implemented in `HarmonicSums`. This solver finds all solutions of holonomic differential equations that can be expressed in terms of iterated integrals over hyperexponential alphabets [9, 12, 13, 23, 33] (with harmonic polylogarithms [34], cyclotomic polylogarithms [1] and iterated integrals over root-valued alphabets [7] as special cases); these solutions are called d'Alembertian solutions [12], in addition for differential equations of order two it finds all solutions that are Liouvillian [10, 30].

*Example 1 (cmAZ)* The following problem, which was already solved in [22], was communicated to us by D. Broadhurst. The goal is to find a differential equation satisfied by

$$Y(h) = \int_0^1 \int_u^1 \frac{1}{\sqrt{uv(1-u)(1-v)(1-uh)(1-(1-v)h)}} dv du. \quad (29)$$

In order to fit (29) to the requirements of the AZ-algorithm we transform it using the substitution  $v \rightarrow u/(1 + (u-1)z)$ , which leads to

$$\int_0^1 \int_0^1 \frac{1}{\sqrt{(1-hu)(z-1)(1+(u-1)z)(h(u-1)(z-1)+z-uz-1)}} dz du. \quad (30)$$

Now we can apply our implementation:

$$\begin{aligned} \text{In[1]:= cmAZ} & \left[ \frac{1}{\sqrt{(1-hu)(z-1)(1+(u-1)z)(h(u-1)(z-1)+z-uz-1)}}, h, \{u, z\}, \right. \\ & \left. \text{AddFactors} \rightarrow \{(1-u)^3(1-z)^6, (1-z)^3\} \right] \\ \text{Out[1]=} & -1 + 2h + 2(1-7h+7h^2)D_h + 6(-1+h)h(-1+2h)D_h^2 + 2(-1+h)^2h^2D_h^3 \end{aligned}$$

Note that in this example the integrand is not vanishing at the integration bounds, still we could derive a homogeneous differential equation, for details we refer to the next session. However, here the right hand side can be computed easily and we find the following differential equation, which is equivalent to the one found in [22]:

$$\left( (h-1)^2h^2D_h^3 + 3(h-1)h(2h-1)D_h^2 + (1-7h+7h^2)D_h + h - \frac{1}{2} \right) Y(h) = \frac{(h^2 + 4h - 4)}{\sqrt{1-h}(2-h)^2}.$$

In a similar way this was already proven by D. van Straten.

## 3.2 Dealing with non-standard boundary conditions

Unfortunately, in many cases the integrand (23) does not vanish at the integration bounds and we end up in a linear differential equation with a non-trivial inhomogeneous part which can be written as a linear combination of integrals with at least one integral operator less. In the following we will deal with non-standard boundary conditions in two different ways, similar to the discrete case of Section 2.

### 3.2.1 Dealing with inhomogeneous differential equations

In the previous section a method that deals with the inhomogeneous recurrences was stated, here we will use similar considerations that will give rise to a recursive method. To be more precise, we consider the integral

$$\mathcal{I}(x) := \int_{u_d}^{o_d} \cdots \int_{u_1}^{o_1} F(x; x_1, \dots, x_d) dx_1 \dots dx_d.$$

Suppose that we found

$$\sum_{i=0}^L e_i(x) D_x^i F(x; x_1, \dots, x_d) = \sum_{i=1}^d D_{x_i} G_i(x; x_1, \dots, x_d) \quad (31)$$

where at least one  $G_i(x; x_1, \dots, x_d)$  does not vanish at the integration limits. By integration with respect to  $x_1, \dots, x_d$  we can deduce that  $\mathcal{I}(x)$  satisfies the inhomogeneous

geneous linear differential equation

$$\begin{aligned} \sum_{i=0}^L e_i(x) D_x^i \mathcal{I}(x) = \\ \sum_{i=1}^d \int_{u_d}^{o_d} \cdots \int_{u_{i-1}}^{o_{i-1}} \int_{u_{i+1}}^{o_{i+1}} \cdots \int_{u_1}^{o_1} O_i(x) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_d \\ - \sum_{i=1}^d \int_{u_d}^{o_d} \cdots \int_{u_{i-1}}^{o_{i-1}} \int_{u_{i+1}}^{o_{i+1}} \cdots \int_{u_1}^{o_1} U_i(x) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_d \end{aligned}$$

with

$$\begin{aligned} U_i(x) &:= G_i(x; x_1, \dots, x_{i-1}, o_i, x_{i+1}, \dots, x_d) \\ O_i(x) &:= G_i(x; x_1, \dots, x_{i-1}, u_i, x_{i+1}, \dots, x_d). \end{aligned}$$

Note that the inhomogeneous part of the above differential equation is a sum of  $2 \cdot d$  integrals of dimension  $d - 1$ , which fit again into the input class of the continuous multiple Almkvist-Zeilberger algorithm. Hence we can apply the algorithms to the  $2 \cdot d$  integrals recursively until we arrive at the base case of one-dimensional integrals for which we have to solve an inhomogeneous differential equation where the inhomogeneous part is free of integrals. Given the solutions for the one-dimensional integrals we can step by step find the solutions of higher dimensional integrals until we finally find the solution for  $\mathcal{I}(x)$  by solving again an inhomogeneous linear differential equation and combining it with the initial conditions. Note that we have to calculate the initial conditions with respect to  $x$  for all the integrals arising in this process.

Summarizing, we use the following strategy (note that we assume that we are able to compute the initial conditions for the arising integrals):

#### Divide and conquer strategy

1. BASE CASE: If  $\mathcal{I}(x)$  has no integration quantifiers, return  $\mathcal{I}(x)$ .
2. DIVIDE: As worked out above, compute a differential equation

$$a_0(x)\mathcal{I}(x) + a_1(x)D_x\mathcal{I}(x) + \cdots + a_d(x)D_x^d\mathcal{I}(x) = h(x) \quad (32)$$

with polynomial coefficients  $a_i(x) \in \mathbb{K}[x]$ ,  $a_m(x) \neq 0$  and the right side  $h(x)$  containing a linear combination of hyperexponential multi-integrals each with less than  $d$  integration quantifiers.

3. CONQUER: Apply the strategy recursively to the simpler integrals in  $h(x)$ . This results in iterated integral expressions  $\tilde{h}(x)$  with

$$\tilde{h}(x) = h(x). \quad (33)$$

If the method fails to find the  $\tilde{h}(x)$  in terms of iterated integral expressions, STOP.



4. COMBINE: Given (32) with (33), compute, if possible,  $\tilde{I}(x)$  in terms of iterated integral expressions such that

$$\tilde{I}(x) = I(x) \quad (34)$$

by solving the differential equation.

This divide and conquer strategy is implemented in the command `cmAZIntegrate` of `MultiIntegrate`.

*Remark 3* We remark that this approach works nicely, if the initial conditions of the integrals in the inhomogeneous part can be calculated efficiently.

*Example 2 (cmAZIntegrate)* We consider the integral

$$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 e^{-x(w_1 w_2 + w_3 w_4)} dw_4 dw_3 dw_2 dw_1 : \quad (35)$$

In[2]:= `cmAZIntegrate[e-x(w1w2+w3w4), x, {{w1, -1, 1}, {w2, -1, 1}, {w3, -1, 1}, {w4, -1, 1}}]`

$$\text{Out[2]} = \frac{8 \left( G\left(\frac{e^{-x}}{x}, \frac{e^{-x}}{x}; x\right) - G\left(\frac{e^{-x}}{x}, \frac{e^{-x}}{x}; x\right) - G\left(\frac{e^{-x}}{x}, \frac{e^{-x}}{x}; x\right) + G\left(\frac{e^{-x}}{x}, \frac{e^{-x}}{x}; x\right) \right)}{x^2}$$

Note that the iterated integrals are defined recursively by

$$G(f_1(\tau), f_2(\tau), \dots, f_k(\tau); x) = \int_0^x f_1(\tau_1) G(f_2(\tau), \dots, f_k(\tau); \tau_1) d\tau_1,$$

with the special case  $G(x) = 1$ , compare *e.g.*, [11].

Here, in a first step the differential equation

$$\begin{aligned} 2f(x) + xD_x f(x) = & \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 e^{x(-w_1 w_2 + w_4)} dw_4 dw_2 dw_1 + \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 e^{-x(w_1 w_2 + w_4)} dw_4 dw_2 dw_1 \\ & + \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 e^{-x(w_2 + w_3 w_4)} dw_4 dw_3 dw_2 + \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 e^{x(w_2 - w_3 w_4)} dw_4 dw_3 dw_2 \end{aligned}$$

is computed. The procedure is applied recursively to all the integrals on the right hand side, which leads to

$$2f(x) + x f'(x) = - \frac{8e^{-x} (e^{2x} - 1) \left( G\left(\frac{e^{-x}}{x}; x\right) - G\left(\frac{e^{-x}}{x}; x\right) \right)}{x^2}$$

Finally, solving this differential equation and combining with initial conditions yields the result.

### 3.2.2 Adapting the ansatz to find homogeneous differential equations

In order to avoid the difficulties of inhomogeneous differential equations we adapt the ansatz. Namely, we can always obtain a homogeneous differential equation of the form (28) by changing (26) to

$$G_i(x; x_1, \dots, x_d) = \overline{H}(x; x_1, \dots, x_d) \cdot r_i(x_1, \dots, x_d) \cdot X_i(x_1, \dots, x_d)(x_i - u_i)(x_i - o_i), \quad (36)$$

i.e., the  $G_i$  are forced to vanish at the integration bounds. Then with this ansatz (27) the underlying linear system turns into

$$\sum_{i=1}^d [D_{x_i} r_i(x_1, \dots, x_d) + q_i(x_1, \dots, x_d)] \cdot X_i(x_1, \dots, x_d)(x_i - u_i)(x_i - o_i) + r_i(x_1, \dots, x_d) \cdot D_{x_i} X_i(x_1, \dots, x_d)(x_i - u_i)(x_i - o_i) = h(x, x_1, \dots, x_d). \quad (37)$$

The general method now is straightforward: Given an integrand of the form (23), we can set  $L = 0$ , look for degree bounds for  $X_i(x, x_1, \dots, x_d)$  and try to find a solution of (37) by coefficient comparison. If we do not find a solution of (37) with not all  $e_i(x)$ 's equal to zero (again homomorphic image testing is used for speedups), we increase  $L$  by one, look for new degree bounds for  $X_i(x, x_1, \dots, x_d)$  and try again to find a solution of (37). Again, if we do not find a solution with not all  $e_i(x)$ 's equal to zero, we increase  $L$  by one and repeat the process.

Once we found a differential equation we can use the differential equation solver implemented in the package HarmonicSums to try to find a closed form solution.

This strategy implemented in the command `cmAZDirectIntegrate` of `MultiIntegrate`.

*Remark 4* The advantage of this approach is, that we do not have to deal with integrals (and initial conditions) recursively, since the differential equation is homogenous, however the additional conditions on the ansatz might increase the order of the differential equation drastically.

*Example 3 (cmAZDirectIntegrate)* We consider again the integral given in (35):

$$\text{In[3]:= cmAZDirectIntegrate}[e^{-x(w_1*w_2+w_3*w_4)}, x, \{\{w_1, -1, 1\}, \{w_2, -1, 1\}, \{w_3, -1, 1\}, \{w_4, -1, 1\}\}]$$

$$\text{Out[3]= } \frac{8 \left( G\left(\frac{e^{-x}}{x}, \frac{e^{-x}}{x}; x\right) - G\left(\frac{e^{-x}}{x}, \frac{e^x}{x}; x\right) - G\left(\frac{e^x}{x}, \frac{e^{-x}}{x}; x\right) + G\left(\frac{e^x}{x}, \frac{e^x}{x}; x\right) \right)}{x^2}$$

Here the differential equation

$$\begin{aligned} 0 = & 32 \left( 4x - 16x^3 + 9x^5 \right) f(x) - 4 \left( 27 - 148x^2 + 598x^4 - 63x^6 \right) f'(x) \\ & - 4 \left( 117x - 568x^3 + 556x^5 - 9x^7 \right) f''(x) - \left( 478x^2 - 2919x^4 + 603x^6 \right) f^{(3)}(x) \\ & - 5 \left( 34x^3 - 247x^5 + 9x^7 \right) f^{(4)}(x) - \left( 23x^4 - 189x^6 \right) f^{(5)}(x) - \left( x^5 - 9x^7 \right) f^{(6)}(x) \end{aligned}$$

is derived. Solving and combing it with the initial condition yields the result given in Out [3].

### 3.3 Computing series expansions of the integrals

Due to time and memory limitations, not finding all solutions of the differential equations or due to missing initial conditions (in full generality) we might fail to process certain integrals using the methods described in the previous subsection. Therefore, inspired by the previous section we are seeking a method which computes  $\varepsilon$ -expansions of integrals of the form (4).

Again we assume that the integral  $\mathcal{I}(\varepsilon, x)$  from (4) has a Laurent expansion in  $\varepsilon$  for  $x \in \mathbb{R}$  with  $x_\alpha < x < x_\beta$  for some  $x_\alpha < 0, x_\beta > 0 \in \mathbb{R}$  and thus it is an analytic function in  $\varepsilon$  throughout an annular region centered by 0 where the pole at  $\varepsilon = 0$  has some order  $K \in \mathbb{Z}$ . Hence we can write it in the form

$$\mathcal{I}(\varepsilon, x) = \sum_{k=-K}^{\infty} \varepsilon^k I_k(x). \quad (38)$$

In the following we try to find the first coefficients  $I_t(x), I_{t+1}(x), \dots, I_u(x)$  in terms of iterated integral expressions of the expansion

$$\mathcal{I}(\varepsilon, x) = I_t(x)\varepsilon^t + I_{t+1}(x)\varepsilon^{t+1} + I_{t+2}(x)\varepsilon^{t+2} + \dots \quad (39)$$

with  $t = -K \in \mathbb{Z}$ . Assume that we managed to compute a differential equation satisfied by  $\mathcal{I}(\varepsilon, x)$  in the form

$$\begin{aligned} a_0(\varepsilon, x)J(\varepsilon, x) + a_1(\varepsilon, x)D_x J(\varepsilon, x) + \dots + a_d(\varepsilon, x)D_x^d J(\varepsilon, x) \\ = h_{-K}(x)\varepsilon^{-K} + h_{-K+1}(x)\varepsilon^{-K+1} + \dots + h_u(x)\varepsilon^u + \dots \end{aligned} \quad (40)$$

In order to find such a differential equation we can use the methods presented in the previous subsections. In the package `HarmonicSums` we implemented an algorithm that tries to find (39), given a differential equation (40) and suitable initial conditions given as power series expansions about  $x = 0$  starting from some  $s \in \mathbb{Z}$ :

$$\begin{aligned}
I_t(x) &= I_{t,s}x^s + I_{t,s+1}x^{s+1} + \cdots + I_{t,s+2}x^{s+d-1} + O(x^{s+d}) \\
I_{t+1}(x) &= I_{t+1,s}x^s + I_{t+1,s+1}x^{s+1} + \cdots + I_{t+1,s+2}x^{s+d-1} + O(x^{s+d}) \\
&\vdots \\
I_u(x) &= I_{u,s}x^s + I_{u,s+1}x^{s+1} + \cdots + I_{u,s+2}x^{s+d-1} + O(x^{s+d})
\end{aligned} \tag{41}$$

In the following we will illustrate the basic calculation steps of this algorithm, which can be considered as the continuous version of the algorithm presented in [18]; see Section 2.3. Inserting the ansatz (39) into (40) yields

$$\begin{aligned}
&a_0(\varepsilon, x) \left[ I_t(x)\varepsilon^t + I_{t+1}(x)\varepsilon^{t+1} + I_{t+2}(x)\varepsilon^{t+2} + \dots \right] + \\
&a_1(\varepsilon, x) \left[ D_x I_t(x)\varepsilon^t + D_x I_{t+1}(x)\varepsilon^{t+1} + D_x I_{t+2}(x)\varepsilon^{t+2} + \dots \right] \\
&+ \cdots + \\
&a_d(\varepsilon, x) \left[ D_x^d I_t(x)\varepsilon^t + D_x^d I_{t+1}(x)\varepsilon^{t+1} + D_x^d I_{t+2}(x)\varepsilon^{t+2} + \dots \right] \\
&= h_t(x)\varepsilon^t + h_{t+1}(x)\varepsilon^{t+1} + \cdots + h_u(x)\varepsilon^u + \dots
\end{aligned} \tag{42}$$

Since two Laurent series agree if they agree coefficient-wise, we obtain the following constraint for  $I_t(x)$  by coefficient comparison:

$$\sum_{k=0}^d a_k(0, x) D_x^k I_t(x) = h_t(x), \tag{43}$$

with the initial condition given in (41). We are now in the position to try to find an explicit representation using HarmonicSums's differential equation solver [9, 10]. We assume that we could find an iterated integral representation  $\tilde{I}_t(x)$  such that  $\tilde{I}_t(x) = I_t(x)$  for all  $x \in (x_\alpha, x_\beta)$ . In order to obtain the next coefficient of the Laurent series in  $\varepsilon$ , we insert  $\tilde{I}_t(x)$  into (42), which yields

$$\begin{aligned}
&a_0(\varepsilon, x) \left[ \tilde{I}_{t+1}(x)\varepsilon^{t+1} + \tilde{I}_{t+2}(x)\varepsilon^{t+2} + \tilde{I}_{t+3}(x)\varepsilon^{t+3} + \dots \right] + \\
&a_1(\varepsilon, x) \left[ D_x \tilde{I}_{t+1}(x)\varepsilon^{t+1} + D_x \tilde{I}_{t+2}(x)\varepsilon^{t+2} + D_x \tilde{I}_{t+3}(x)\varepsilon^{t+3} + \dots \right] \\
&+ \cdots + \\
&a_d(\varepsilon, x) \left[ D_x^d \tilde{I}_{t+1}(x)\varepsilon^{t+1} + D_x^d \tilde{I}_{t+2}(x)\varepsilon^{t+2} + D_x^d \tilde{I}_{t+3}(x)\varepsilon^{t+3} + \dots \right] \\
&= \tilde{h}_{t+1}(x)\varepsilon^{t+1} + \tilde{h}_{t+2}(x)\varepsilon^{t+2} + \cdots + \tilde{h}_u(x)\varepsilon^u + \dots
\end{aligned} \tag{44}$$

with  $\tilde{h}_i(x) = h_i(x) - g_i(x)$ , where the  $g_i(x)$  satisfy

$$\begin{aligned}
&a_0(\varepsilon, x)\tilde{I}_t(x) + a_1(\varepsilon, x)D_x \tilde{I}_t(x) + \cdots + a_d(\varepsilon, x)D_x^d \tilde{I}_t(x) \\
&= h_t(x)\varepsilon^t + g_{t+1}(x)\varepsilon^{t+1} + \cdots + g_u(x)\varepsilon^u + \dots
\end{aligned}$$

Now we repeat the above procedure: by coefficient comparison we obtain the following constraint for  $I_{t+1}(x)$ :

$$\sum_{k=0}^d a_k(0, x) D_x^k I_{t+1}(x) = \tilde{h}_{t+1}(x). \quad (45)$$

Assuming that we can find a solution  $\tilde{I}_{t+1}(x)$  of (45) in terms of iterated integrals that satisfy the initial condition from (41) such that  $\tilde{I}_{t+1}(x) = I_{t+1}(x)$  for all  $x \in (x_\alpha, x_\beta)$  we can update the ansatz (44):

$$\begin{aligned} & a_0(\varepsilon, x) \left[ I_{t+2}(x) \varepsilon^{t+2} + I_{t+3}(x) \varepsilon^{t+3} + I_{t+4}(x) \varepsilon^{t+4} + \dots \right] + \\ & a_1(\varepsilon, x) \left[ D_x I_{t+2}(x) \varepsilon^{t+2} + D_x I_{t+3}(x) \varepsilon^{t+3} + D_x I_{t+4}(x) \varepsilon^{t+4} + \dots \right] \\ & + \dots + \\ & a_d(\varepsilon, x) \left[ D_x^d I_{t+2}(x) \varepsilon^{t+2} + D_x^d I_{t+3}(x) \varepsilon^{t+3} + D_x^d I_{t+4}(x) \varepsilon^{t+4} + \dots \right] \\ & = \tilde{h}_{t+2}(x) \varepsilon^{t+2} + \tilde{h}_{t+3}(x) \varepsilon^{t+3} + \dots + \tilde{h}_u(x) \varepsilon^u + \dots \end{aligned} \quad (46)$$

with  $\tilde{h}_i(x) = \tilde{h}_i(x) - \tilde{g}_i(x)$ , where the  $\tilde{g}_i(x)$  satisfy

$$\begin{aligned} & a_0(\varepsilon, x) \tilde{I}_{t+1}(x) + a_1(\varepsilon, x) D_x \tilde{I}_{t+1}(x) + \dots + a_d(\varepsilon, x) D_x^d \tilde{I}_{t+1}(x) \\ & = \tilde{h}_{t+1}(x) \varepsilon^t + \tilde{g}_{t+2}(x) \varepsilon^{t+2} + \dots + \tilde{g}_u(x) \varepsilon^u + \dots \end{aligned}$$

We can repeat this process as long as we can compute solutions and as long as needed. The illustrated calculation steps can be summarized with the following theorem.

**Theorem 1** *Suppose we are given a linear differential equation*

$$\begin{aligned} & a_0(\varepsilon, x) J(\varepsilon, x) + a_1(\varepsilon, x) D_x J(\varepsilon, x) + \dots + a_d(\varepsilon, x) D_x^d J(\varepsilon, x) \\ & = h_k(x) \varepsilon^k + h_{k+1}(x) \varepsilon^{k+1} + \dots + h_u(x) \varepsilon^u + \dots \end{aligned}$$

*of order  $d$  where the  $a_i(\varepsilon, x)$  are polynomials in  $x$  and  $\varepsilon$  and where the inhomogeneous part can be expanded in  $\varepsilon$  up to order  $u$  in terms of expressions in iterated integrals over hyperexponential alphabets. Consider a function which has a Laurent series expansion*

$$J(\varepsilon, x) = F_k(x) \varepsilon^k + F_{k+1}(x) \varepsilon^{k+1} + \dots$$

*and which is a solution of the given differential equation for all  $x \in \mathbb{R}$  with  $x_\alpha < x < x_\beta$  for some  $x_\alpha < 0, x_\beta > 0 \in \mathbb{R}$ . Then together with the initial conditions*

$$F_j(x) = F_{j,s} x^s + F_{j,s+1} x^{s+1} + \dots + F_{j,s+2} x^{s+d-1} + O(x^{s+d})$$

*with  $k \leq j \leq u$ , all  $F_k(x), \dots, F_u(x)$  with  $x_\alpha < x < x_\beta$  can be computed in terms of expressions in iterated integrals over hyperexponential alphabets provided that the*

values  $h_j(x)$  for all  $j$  with  $k \leq i \leq u$  and  $x_\alpha < x < x_\beta$  can be computed in terms of expressions in iterated integrals over hyperexponential alphabets.

This algorithm is implemented in the package `HarmonicSums` and with this implementation in hand we can try to find Laurent series solutions of integrals of the form (4). Let  $I(\varepsilon, n)$  be a multi-integral of the form (4) and assume that  $I(\varepsilon, x)$  has a series expansion (39) for all  $x \in \mathbb{R}$  with  $x_\alpha < x < x_\beta$  for some  $x_\alpha < 0, x_\beta > 0 \in \mathbb{R}$ . If we succeed in finding a homogeneous differential equation, for instance by using the method from Section 3.2.1 we can directly apply the Laurent series differential equation solver, supposing that we can handle the initial conditions.

This strategy is implemented in the command `cmAZExpandedDirectIntegrate` of `MultiIntegrate`.

*Example 4 (cmAZExpandedDirectIntegrate)* We consider the integral

$$I(\varepsilon, w) = \int_0^1 \int_0^1 \int_0^1 e^{xyw} ((1-w)x(1-y))^{\frac{\varepsilon}{2}} ((1-w)y(1-x)z(1-z)) dz dy dx \quad (47)$$

with the given initial condition

$$I(\varepsilon, w) = \frac{8}{3(2+\varepsilon)^2(4+\varepsilon)^2} - \frac{4w(28+\varepsilon(12+\varepsilon))}{3(2+\varepsilon)(4+\varepsilon)^2(6+\varepsilon)^2} + \underbrace{\frac{w^2(-1664+\varepsilon(12+\varepsilon(12+\varepsilon))(72+\varepsilon(16+\varepsilon)))}{3(2+\varepsilon)(4+\varepsilon)^2(6+\varepsilon)^2(8+\varepsilon)^2}}_{init:=} + O(w^3).$$

We want to find the first two terms of the  $\varepsilon$ -expansion of  $I(\varepsilon, w)$ , i.e., we want to compute  $I_0(w)$  and  $I_1(w)$  such that  $I(\varepsilon, w) = I_0(w) + \varepsilon I_1(w) + O(\varepsilon^2)$ . This can be achieved by using our implementation:

`in[4]:= cmAZExpandedDirectIntegrate[I(ε, w), w, {ε, 0, 1}, {{x, 0, 1}, {y, 0, 1}, {z, 0, 1}}, InitValues → init]`

$$\text{Out[4]= } \left\{ \left\{ \frac{1}{6} - \frac{1}{6w} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{6w^2} + \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{6w}, -\frac{1}{12w^2} + \frac{\varepsilon^w}{12w^2} + \frac{1}{12w} - \frac{\varepsilon^w}{12w} \right. \right. \\ \left. - \frac{G\left(\frac{1}{1-\tau}; w\right)}{12} + \frac{G\left(\frac{1}{1-\tau}; w\right)}{12w} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{12} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w^2} + \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{6w} \right. \\ \left. + \frac{G\left(\frac{1}{1-\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1}{1-\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w} - \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w^2} + \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w} \right\}$$

$$\begin{aligned}
& + \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w} + \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1}{1-\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1}{1-\tau}; w\right)}{12w} \\
& + \left. \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w} \right\}, \{0, 1\} \}
\end{aligned}$$

Of course we can again think of a recursive method to compute the first coefficients, say  $F_t(n), \dots, F_u(n)$  of (39). Note that we have the same advantages and disadvantages as mentioned for the recursive method in Section 3.2.2, but if we assume that we can handle the initial conditions we can use to following strategy.

### Divide and conquer strategy

1. BASE CASE: If  $I(\varepsilon, x)$  has no integration quantifiers, compute the expansion by standard methods.
2. DIVIDE: As worked out before, compute a differential equation

$$a_0(\varepsilon, x)I(\varepsilon, x) + \dots + a_d(\varepsilon, x)D_x^d I(\varepsilon, x) = h(\varepsilon, x) \quad (48)$$

with polynomial coefficients  $a_i(\varepsilon, x) \in \mathbb{K}[\varepsilon, x]$ ,  $a_m(\varepsilon, x) \neq 0$  and the right side  $h(\varepsilon, x)$  containing a linear combination of hyperexponential multi-integrals each with less than  $d$  integration quantifiers.

3. CONQUER: Apply the strategy recursively to the simpler integrals in  $h(\varepsilon, x)$ . This results in an expansion of the form

$$h(\varepsilon, x) = h_t(x) + h_1(x)\varepsilon + \dots + h_u(x)\varepsilon^u + O(\varepsilon^{u+1}); \quad (49)$$

if the method fails to find the  $h_t(x), \dots, h_u(x)$  in terms of iterated integral expressions, STOP.

4. COMBINE: Given (48) with (49), compute, if possible, the  $F_t(n), \dots, F_u(n)$  of (39) in terms of iterated integral expressions by using `HarmoniCSums`.

This divide and conquer strategy is implemented in the command `cmAZExpandedIntegrate` of `MultiIntegrate`.

*Example 5 (cmAZExpandedIntegrate)* Again we consider the integral given in (47) with the same initial condition. In order to compute the first two terms of the  $\varepsilon$ -expansion of  $I(\varepsilon, w)$ , we can also use the following function call:

```

In[5]:= cmAZExpandedIntegrate[I(ε, w), w, {ε, 0, 1}, {{x, 0, 1}, {y, 0, 1}, {z, 0, 1}},
      InitValues → init]

```

$$\text{Out[5]=} \left\{ \left\{ \frac{1}{6} - \frac{1}{6w} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{6w^2} + \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{6w}, -\frac{1}{12w^2} + \frac{\varepsilon^w}{12w^2} + \frac{1}{12w} - \frac{\varepsilon^w}{12w} \right. \right. \\ - \frac{G\left(\frac{1}{1-\tau}; w\right)}{12} + \frac{G\left(\frac{1}{1-\tau}; w\right)}{12w} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{12} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w^2} + \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}; w\right)}{6w} \\ + \frac{G\left(\frac{1}{1-\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1}{1-\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w} - \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w^2} + \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w} \\ + \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1}{\tau}, \frac{1-\varepsilon^\tau}{\tau}; w\right)}{12w} + \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1}{1-\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1}{1-\tau}; w\right)}{12w} \\ \left. \left. + \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w^2} - \frac{G\left(\frac{1-\varepsilon^\tau}{\tau}, \frac{1-\varepsilon^{-\tau}}{\tau}; w\right)}{12w} \right\}, \{0, 1\} \right\}$$

## Conclusion

In this paper we summarize the theoretical background of our package MultiIntegrate which can be downloaded at <https://risc.jku.at/software> and which provides several methods to deal with multiple integrals over hyperexponential integrands.

**Acknowledgements** This work was supported by the Austrian Science Fund (FWF) grant SFB F50 (F5009-N15) and by the bilateral project WTZ BG 03/2019 (KP-06-Austria/8/2019), funded by OeAD (Austria) and Bulgarian National Science Fund. The author would like to thank C. Schneider for useful discussions.

## References

1. J. Ablinger, J. Blümlein and C. Schneider. Harmonic Sums and Polylogarithms Generated by Cyclotomic Polynomials. *Journal of Mathematical Physics*, 52(10):102301, 2011.
2. J. Ablinger. Computer Algebra Algorithms for Special Functions in Particle Physics. PhD thesis, J. Kepler University Linz, 2012.
3. J. Ablinger, J. Blümlein, M. Round and C. Schneider, Advanced Computer Algebra Algorithms for the Expansion of Feynman Integrals. *PoS LL2012*, 050, 2012.
4. J. Ablinger, J. Blümlein and C. Schneider. Analytic and Algorithmic Aspects of Generalized Harmonic Sums and Polylogarithms. *Journal of Mathematical Physics*, 54(8):082301, 2013.
5. J. Ablinger. The package HarmonicSums: Computer Algebra and Analytic aspects of Nested Sums. In *PoS LL2014*, 019, 2014.
6. J. Ablinger, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel, M. Round, C. Schneider and F. Wißbrock. The Transition Matrix Element  $A_{gq}(N)$  of the Variable Flavor Number Scheme at  $O(\alpha_s^3)$ . *Nuclear Physics B*, 882:263-288, 2014.
7. J. Ablinger, J. Blümlein, C. G. Raab and C. Schneider. Iterated Binomial Sums and their Associated Iterated Integrals. *Journal of Mathematical Physics*, 55(11):112301, 2014.



8. J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel and C. Schneider. Calculating Three Loop Ladder and V-Topologies for Massive Operator Matrix Elements by Computer Algebra. *Computer Physics Communications*, 202:33–112, 2016.
9. J. Ablinger. Inverse Mellin Transform of Holonomic Sequences. In *PoS LL2016*, 067, 2016.
10. J. Ablinger. Computing the Inverse Mellin Transform of Holonomic Sequences using Kovacic’s Algorithm. In *PoS RADCOR2017*, 069, 2018.
11. J. Ablinger. Discovering and Proving Infinite Pochhammer Sum Identities. *Experimental Mathematics*, 2019.
12. S. Abramov and M. Petkovšek. D’Alembertian solutions of linear differential and difference equations. In J. von zur Gathen, editor, *Proceedings of ISSAC’94*, pages 169–174, 1994.
13. S. Abramov and E. Zima. D’Alembertian Solutions of Inhomogeneous Linear Equations (differential, difference, and some other). In Y. N. Lakshman, editor, *Proceedings of ISSAC’96*, pages 232–240, 1996.
14. M. Apagodu and D. Zeilberger. Multi-variable Zeilberger and Almkvist-Zeilberger algorithms and the sharpening of Wilf-Zeilberger theory. *Advances in Applied Mathematics*, 37(2):139–152, 2006.
15. I. Bierenbaum, J. Blümlein, and S. Klein. Mellin moments of the  $o(\alpha_s^3)$  heavy flavor contributions to unpolarized deep-inelastic scattering at  $q^2 \gg m^2$  and anomalous dimensions. *Nuclear Physics B*, 820(1):417–482, 2009.
16. J. Blümlein and S. Kurth. Harmonic sums and Mellin transforms up to two-loop order. *Physical Review D*, 60(1):014018, 1999.
17. J. Blümlein, M. Kauers, S. Klein, and C. Schneider. Determining the closed forms of the  $O(\alpha_s^3)$  anomalous dimensions and Wilson coefficients from Mellin moments by means of computer algebra. *Computer Physics Communications*, 180(11):2143–2165, 2009.
18. J. Blümlein, S. Klein, C. Schneider, and F. Stan. A symbolic summation approach to Feynman integral calculus. *Journal of Symbolic Computation*, 47(10):1267–1289, 2011.
19. J. Blümlein, A. Hasselhuhn, and C. Schneider. Evaluation of multi-sums for large scale problems. In *PoS RADCOR2011*, 032, 2012.
20. J. Blümlein, M. Round, and C. Schneider. Refined Holonomic Summation Algorithms in Particle Physics. In *Advances in Computer Algebra. WWCA 2016*, 226:51–91, 2018.
21. C. Bogner and S. Weinzierl. Feynman graph polynomials. *International Journal of Modern Physics A*, 25:2585–2618, 2010.
22. D. Broadhurst and W. Zudilin. A magnetic double integral. *Journal of the Australian Mathematical Society*, 1:9–25, 2019.
23. M. Bronstein. Linear Ordinary Differential Equations: breaking through the order 2 barrier. In *Proceedings of ISSAC’92*, pages 42–48, 1992.
24. S. Chen and M. Kauers. Trading order for degree in creative telescoping. *Journal of Symbolic Computation*, 47(8):968–995, 2012.
25. S. Chen and M. Kauers. Order-degree curves for hypergeometric creative telescoping. In *Proceedings of ISSAC’12*, pages 122–129, 2012.
26. S. Chen, M. Kauers, and C. Koutschan. A generalized apagodu-zeilberger algorithm. In *Proceedings of ISSAC’14*, pages 107–114, 2014.
27. Y. Frishman. Operator products at almost light like distances. *Annals of Physics*, 66:373–389, 1971.
28. M. Y. Kalmykov and O. Veretin. Single scale diagrams and multiple binomial sums. *Physics Letters B*, 483:315–323, 2000.
29. C. Koutschan. A fast approach to creative telescoping. *Mathematics in Computer Science*, 4(2–3):259–266, 2010.
30. J.J. Kovacic, An algorithm for solving second order linear homogeneous differential equations. *Journal of Symbolic Computation*, 2, 1986.
31. S.-O. Moch, P. Uwer and S. Weinzierl. Nested sums, expansion of transcendental functions, and multiscale multiloop integrals. *Journal of Mathematical Physics*, 43(6):3363–3386, 2002.
32. M. Mohammed and D. Zeilberger. Sharp upper bounds for the orders of the recurrences outputted by the Zeilberger and q-Zeilberger algorithms. *Journal of Symbolic Computation*, 39(2):201–207, 2005.

33. M. Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, 14:243–264, 1992.
34. E. Remiddi and J.A.M. Vermaseren. Harmonic polylogarithms. *International Journal of Modern Physics A*, 15:725–754, 2000.
35. C. Schneider. *Symbolic summation in difference fields*. PhD thesis, RISC, J. Kepler University Linz, May 2001.
36. C. Schneider. A new Sigma approach to multi-summation. *Advances in Applied Mathematics*, 34:740–767, 2005.
37. C. Schneider. Solving parameterized linear difference equations in terms of indefinite nested sums and products. *Journal of Difference Equations and Applications*, 11(9):799–821, 2006.
38. C. Schneider. Symbolic summation assists combinatorics. *Séminaire Lotharingien de Combinatoire*, 56:1–36, 2007.
39. C. Schneider. Simplifying Multiple Sums in Difference Fields. In *Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions* Texts & Monographs in Symbolic Computation eds. C. Schneider and J. Blümlein (Springer, Wien, 2013), 325–360, 2013.
40. C. Schneider. Modern Summation Methods for Loop Integrals in Quantum Field Theory: The Packages Sigma, EvaluateMultiSums and SumProduction. *Journal of Physics: Conference Series*, 523:012037, 2014.
41. J.A.M. Vermaseren. Harmonic sums, Mellin transforms and Integrals. *International Journal of Modern Physics A*, 14(13):2037–2076, 1999.
42. K. Wegschaider. Computer generated proofs of binomial multi-sum identities. Master’s thesis, RISC, J. Kepler University, May 1997.
43. S. Weinzierl. Feynman graphs. In *Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions*, Texts & Monographs in Symbolic Computation eds. C. Schneider and J. Blümlein (Springer, Wien, 2013), 381–406, 2013.
44. H. S. Wilf and D. Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and “ $q$ ”) multisum/integral identities. *Inventiones mathematicae*, 108:575–633, 1992.