

DESY 16-160, DO-TH 16-19,

arXiv:1608.05376 [cs.SC]

## Algorithms to solve coupled systems of differential equations in terms of power series\*

---

**Jakob Ablinger and Carsten Schneider<sup>†</sup>***Research Institute for Symbolic Computation (RISC)**Johannes Kepler University, Altenbergerstraße 69, A-4040 Linz, Austria**E-mail: Jakob Ablinger@risc.jku.at, Carsten.Schneider@risc.jku.at***Arnd Behring, Johannes Blümlein and Abilio de Freitas***Deutsches Elektronen-Synchrotron, DESY,**Platanenallee 6, D-15738 Zeuthen, Germany**E-mail: johannes.bluemlein@desy.de, abilio.de.freitas@desy.de,  
arnd.behring@desy.de*

Using integration by parts relations, Feynman integrals can be represented in terms of coupled systems of differential equations. In the following we suppose that the unknown Feynman integrals can be given in power series representations, and that sufficiently many initial values of the integrals are given. Then there exist algorithms that decide constructively if the coefficients of their power series representations can be given within the class of nested sums over hypergeometric products. In this article we will work out the calculation steps that solve this problem. First, we will present a successful tactic that has been applied recently to challenging problems coming from massive 3-loop Feynman integrals. Here our main tool is to solve scalar linear recurrences within the class of nested sums over hypergeometric products. Second, we will present a new variation of this tactic which relies on more involved summation technologies but succeeds in reducing the problem to solve scalar recurrences with lower recurrence orders. The article will work out the different challenges of this new tactic and demonstrates how they can be treated efficiently with our existing summation technologies.

*Loops and Legs in Quantum Field Theory**24-29 April 2016**Leipzig, Germany*

---

\*This work was supported in part by the Austrian Science Fund (FWF) grant SFB F50 (F5009-N15) and the European Commission through contract PITN-GA-2012-316704 (HIGGSTOOLS).

<sup>†</sup>Speaker.

## 1. Introduction

Coupled systems of linear differential equations arise frequently in intermediate calculations when one tries to tackle massless or massive three-loop Feynman integrals. Namely, using integration by parts (IBP) methods<sup>1</sup> [3] one can determine such a coupled system and together with the initial values one obtains a complete description of the involved Feynman integrals. Various techniques [4] have been elaborated to extract relevant information from these systems that sheds light on the involved Feynman integrals. For heuristic methods to find closed form solutions of such systems, in case the solutions are given by iterative integrals, see for instance [5]. A completely algorithmic approach has been worked out recently in [6–8] if the unknown functions  $\hat{I}_i(x)$  with  $1 \leq i \leq r$  of the coupled system admit power series representations<sup>2</sup>

$$\hat{I}_i(x) = \sum_{N=0}^{\infty} I_i(N)x^N. \quad (1.1)$$

Then under the assumption that sufficiently many initial values  $I_i(v)$  at nonnegative integers  $v \in \mathbb{N}$  are available, we can utilize our summation machinery that decides constructively if the  $I_i(N)$  can be written in terms of nested sums over hypergeometric products. This means that they can be composed by elements from the rational function field  $\mathbb{K}(N)$ , the three operations  $(+, -, \cdot)$ , hypergeometric products of the form  $\prod_{k=l}^N h(k)$  with  $l \in \mathbb{N}$  and  $h(k)$  being a rational function in  $k$  and being free of  $N$ , and sums of the form  $\sum_{k=l}^N h(k)$  with  $l \in \mathbb{N}$  and with  $h(k)$  being a nested hypergeometric sum expression w.r.t.  $k$  and being free of  $N$ . This class of special functions covers as special cases harmonic sums [9], generalized harmonic sums [10, 11], cyclotomic harmonic sums [12] or nested binomial sums [13, 14]; for surveys on these quantities see e.g. [15].

The central step of our procedure [6, 7] is to translate the problem of solving a coupled system of differential equations to the problem of solving scalar recurrences that depend only on one of the unknown functions  $I_i(N)$ . To obtain a complete algorithm we utilize uncoupling algorithms [16, 17], recurrence solving algorithms [18–20], and simplifiers of the solutions by means of symbolic summation [21] based on difference field [22] and difference ring [23] theory. This successful interplay of the underlying summation tools (uncoupling, recurrence solving, and indefinite summation) has been implemented within the package `SolveCoupledSystem` [8] that relies on the summation packages `Sigma` and `SumProduction` [21] (concerning the summation tools) and the the package `OreSys` [24] (executing Zürcher’s uncoupling algorithm [16]). The package `SolveCoupledSystem` has been applied successfully for various challenging 3-loop calculations [25]. These heavy calculations were possible by incorporating besides `Sigma` also the package `HarmonicSums`: whenever possible, we employed its algorithms that are tuned for harmonic sums, cyclotomic sums, generalized harmonic sums and nested binomial sums.

Still, in recent calculations two serious problems arose. First, in some instances we obtained coupled systems of difference equations that we failed to uncouple due to restricted time and memory resources. Second, we succeeded in uncoupling the system but derived scalar recurrences whose orders were very large. We emphasize that `Sigma` was still capable of solving these large

<sup>1</sup>In our calculations we used heavily the C++ program `Reduze 2` [1] based on Laporta’s algorithm [2].

<sup>2</sup>In addition there may be trailing terms such as  $\ln^k(x)$  for each of these power series and one may shift  $N$  by a finite number of integer values.

recurrences (in [26] we solved gigantic recurrences up to order 35) but finding 16 or more initial values to combine the solutions accordingly was the bottleneck: extracting such a large amount of initial values from the given Feynman integrals was out of the scope of the existing algorithms, like exploiting the  $\alpha$ -parameterization of the integrals [27].<sup>3</sup> Based on our existing algorithmic machinery (see Subsection 2.1 below) we will present a new tactic in Subsection 2.2 that has been implemented within the package `SolveCoupledSystem` and that overcomes the problems described above. A conclusion concerning future calculations that will rely on this new machinery will be given in Section 3. The present algorithm also decides, whether or not a system can be solved in terms of iterative integrals over whatsoever alphabet or not, and allows therefore to single out systems which are not first order reducible requiring other techniques of solution, cf. [29].

## 2. Two basic strategies

Suppose we are given a coupled system of differential equations coming from IBP-methods [1–3]. More precisely, we are given a matrix  $\hat{A}$  of dimension  $r \times r$  with entries from the rational function field  $\mathbb{K}(x)$  and we are given a vector  $\hat{\mathbf{b}}(x) = (\hat{b}_1(x), \dots, \hat{b}_r(x))$  of length  $r$  where each entry is given in terms of a linear combination of master integrals with coefficients from  $\mathbb{K}(x)$ . Then, given sufficiently many initial values we seek for the uniquely determined solution  $\hat{\mathbf{I}}(x) = (\hat{I}_1(x), \dots, \hat{I}_r(x))$  of the system

$$D_x \hat{\mathbf{I}}(x) = \hat{A} \hat{\mathbf{I}}(x) + \hat{\mathbf{b}}(x) \quad (2.1)$$

with  $D_x \hat{\mathbf{I}}(x) = (D_x \hat{I}_1(x), \dots, D_x \hat{I}_r(x))$ , where  $D_x$  is the differential operator w.r.t.  $x$  acting on functions in  $x$ .

In our setting we may assume that the master integrals arising in the  $\hat{b}_i(x)$  have a power series representation, where the coefficients are already computed. Within our calculations in [25] this has been accomplished with

1. the package `EvaluateMultiSums` [21] relying on the summation package `Sigma` and the special functions package `HarmonicSums` [30] (see also [9–14]);
2. the package `MultiIntegrate` [31] being a specially tuned version of the multivariate Almkvist–Zeilberger algorithm;
3. the package `SolveCoupledSystem` containing the algorithms that will be explained below – here the package has been applied to systems which do not depend on the functions given in  $\mathbf{I}(x)$ .

In particular, we assume that the coefficients are given in terms of nested sums over hypergeometric products. This enables one to write also the  $\hat{b}_i(x)$  in a power series representation

$$\hat{b}_i(x) = \sum_{N=0}^{\infty} b_i(N) x^N \quad (2.2)$$

---

<sup>3</sup>Yet one might think of other methods to obtain these initial values. It is, however, clear that the calculation of finite Mellin moments of complicated diagrams in general constitutes no simple task either, cf. [28], for very high moments.

whose coefficients  $b_i(N)$  are given in terms of nested sums over hypergeometric products. Furthermore, we assume that also the unknown master integrals  $\hat{I}_i$  are analytic, i.e., there exist the power series representations (1.1).

Under these assumptions we seek for the coefficients  $I_1(N), \dots, I_r(N)$  in closed form. This means that we look for representations in terms of nested sums over hypergeometric products whenever such a representation is possible.

In Subsection 2.1 we will recall our first algorithm [6–8] that solves this problem and that has been successfully applied to various challenging massive 3-loop Feynman integrals [25]. In this approach we will address some efficiency problems that have occurred recently. By a variation of the employed building blocks, we will present a new tactic that overcomes these problems in Subsection 2.2.

## 2.1 Tactic 1: Uncoupling a system of difference equations

In the first approach the system of differential equations (2.1) with the unknown functions  $\hat{I}_i(x)$ , which have power series representations (1.1), is translated to a system of difference equations for the corresponding coefficients  $I_i(N)$ . Afterwards one applies computer algebra algorithms in order to find a symbolic representation of the  $I_i(N)$  in terms of nested sums over hypergeometric products. More precisely, we execute the following steps.

- (1) We utilize that an analytic function  $\sum_{N=0}^{\infty} h(N)x^N$  with a certain convergence disc fulfills

$$D_x \sum_{N=0}^{\infty} h(N)x^N = \sum_{N=1}^{\infty} N h(N)x^{N-1}; \quad (2.3)$$

similarly, if we argue in the formal power series setting, the operator  $D_x$  is just defined in this way. Plugging the already computed series expansions of  $\hat{b}_i(x)$  and the Ansatz (1.1) into (2.1) (note that in general we can start with a coupled system of differential equations of higher-order; see [8]) and doing coefficient comparison w.r.t.  $x^N$  yields a coupled system of linear difference equations, say of order  $m \geq 0$ , which is of the form

$$A_m \mathbf{I}(N+m) + A_{m-1} \mathbf{I}(N+m-1) + \dots + A_0 \mathbf{I}(N) = \mathbf{b}(N), \quad (2.4)$$

with  $\mathbf{I}(N+l) = (I_1(N+l), \dots, I_r(N+l))$  for  $l \in \mathbb{N}$  and where the matrices  $A_m, \dots, A_0$  have dimension  $r \times r$  with entries from  $\mathbb{K}(n)$ . Ideally, one should solve this coupled system directly. For rational solutions this problem is solved in [32]; for first steps towards the more general setting of  $\Pi\Sigma$ -fields we refer to [33]. So far, there are no algorithms available to solve such systems directly in terms of nested sums over hypergeometric products. Therefore we will proceed as follows.

- (2) We uncouple the system such that some of the unknown coefficients  $I_i(N)$  of the power series are solutions of a scalar linear recurrence (which does not depend on other unknown functions) and where the remaining unknown coefficients can be given as a linear combination of the coefficients that are determined by the solutions of the scalar recurrences. To accomplish this task, we can apply the following three sub-steps.

(a) First, we transform the system (2.4) of order  $m$  to a first-order system as follows. For  $1 \leq i \leq r$ , let  $n_i \in \mathbb{N}$  be the maximal value such that  $I_i(N + n_i)$  occurs in (2.4). Then one can introduce auxiliary-functions  $y_{i,j}(N)$  with  $1 \leq j < n_i$  and  $1 \leq i \leq r$  where  $I_i(N + j)$  is rephrased by  $y_{i,j}(N)$  in (2.4). Introducing in addition the relations  $y_{i,j}(N + 1) = y_{i,j+1}(N)$  and replacing  $y_{i,n_i}(N)$  by  $y_{i,n_i-1}(N + 1)$  in (2.4) one arrives at a first-order system of difference equations. In other words, solving this first-order system with the unknown  $y_{i,j}(N)$  yields immediately the solutions for the  $I_i(N)$ . For simplicity, we write again  $I_i(N)$  instead of  $y_i(N)$ , and suppose that we are given the system

$$\mathbf{I}(N + 1) = A\mathbf{I}(N) + \mathbf{b}(N), \quad (2.5)$$

where  $A$  is an  $r \times r$  matrix and  $\mathbf{b}(N) = (b_1(N), \dots, b_r(N))$  is a vector where  $b_i(N)$  are expression in terms of nested sums over hypergeometric products.

(b) If  $A$  is an invertible matrix we skip this step. Otherwise, we proceed as follows. One can perform simple row and column operations in order get again a system of this form with dimension  $r' < r$  such that the  $r' \times r'$  matrix is invertible and such that we are given linear combinations of the  $r'$  unknown integrals which produce the desired integrals  $I_i(N)$  for  $1 \leq i \leq r$ . Hence, solving this reduced system provides the solution of the original system (2.5). For further considerations we will continue to work with (2.5) where  $A$  is invertible.

(c) Finally, we use Zürcher's algorithm [16] that is available in the package `OreSys` [24] in order to uncouple this system. In general, one obtains  $n$  ( $1 \leq n \leq r$ ) scalar recurrences, namely, for  $1 \leq i \leq n$  we get the recurrences

$$a_{i,i}(N)I_i(N + m_i) + a_{i,m_i-1}(N)I_i(N + m_i - 1) + \dots + a_{i,0}I_i(N) = f_i(N) \quad (2.6)$$

of orders  $m_i$  where the  $a_{i,j}(N)$  are from  $\mathbb{K}(N)$  and the  $f_i(N)$  are given as a linear combination of the  $b_1(N), \dots, b_r(N)$  over  $\mathbb{K}(N)$  with possible shifts in  $N$ . Since the  $b_i(N)$  are given in terms of nested sums over hypergeometric products, also the  $f_i(N)$  can be given in terms of nested sums over hypergeometric products. In addition, we get explicit linear combinations of the  $I_1(N), \dots, I_n(N)$  and the  $b_1(N), \dots, b_r(N)$  with possible shifts in  $N$  that produce the remaining integrals  $I_{n+1}(N), \dots, I_m(N)$ . More generally, a subset  $M$  of  $\{I_1, \dots, I_m\}$  can be described by scalar recurrences and the complementary set can be represented by a linear combination of the functions from  $M$ ; after reordering, we may suppose that  $M = \{I_1, \dots, I_n\}$ .

(3) Finally, we aim at deciding algorithmically if the coefficients  $I_i(N)$  of the integrals (1.1) can be represented in terms of nested sums over hypergeometric products. In this regard, we have to emphasize that the expressions  $b_i(N)$  of  $\mathbf{b}(N)$  arising in (2.5) depend usually on an extra parameter  $\varepsilon$  coming from the analytic continuation of the space-time dimension  $D = \varepsilon + 4$ . Similarly, the matrix  $A$  in (2.5) depends on this dimensional parameter  $\varepsilon$ . Thus we usually are given the rational function field  $\mathbb{K} = \mathbb{K}'(\varepsilon)$  where  $\mathbb{K}'$  is a subfield of  $\mathbb{K}$  containing  $\mathbb{Q}$ . Further, there is usually no hope to represent the  $b_i(N)$  and also the the integrals  $I_i(N)$  in terms of nested sums over hypergeometric products where  $\varepsilon$  occurs inside of the sums and products. However, in most applications, there is such a representation, if one considers the functions in its  $\varepsilon$ -expansion [20, 34]. More precisely, we assume that the  $b_i(N)$  are given in the form

$$b_i(N) = b_{i,o}(N)\varepsilon^o + b_{i,o+1}(N)\varepsilon^{o+1} + b_{i,o+2}(N)\varepsilon^{o+2} + \dots, \quad (2.7)$$

where the first coefficients  $b_{i,j}(N)$  are given in terms of nested sums over hypergeometric products (which are free of  $\varepsilon$ ). Further, one is not interested in an expression of the  $I_i(N)$ , where the nested sums and products depend on  $\varepsilon$ , but one is interested in the first  $l_i$  coefficients of the  $\varepsilon$ -expansion

$$I_i(N) = I_{i,o}(N)\varepsilon^o + I_{i,o+1}(N)\varepsilon^{o+1} + \dots I_{i,o}(N)\varepsilon^{o+l_i},$$

where  $I_{i,j}(N)$  is given in terms of nested sums over hypergeometric products (which are free of  $\varepsilon$ ). In our 3-loop calculations we usually have  $o = -3$ . More precisely, suppose that we are given the initial values

$$I_i(\mathbf{v}) = I_{i,o}(\mathbf{v})\varepsilon^o + I_{i,o+1}(\mathbf{v})\varepsilon^{o+1} + \dots I_{i,o+l_i}(\mathbf{v})\varepsilon^{o+l_i} + \dots \quad (2.8)$$

for<sup>4</sup>  $\mathbf{v} = 1, \dots, m_i$ . Then we can decide algorithmically if the coefficients  $I_{i,j}(N)$  can be represented in terms of nested sums over hypergeometric products by executing the following sub-steps.

- (a) Note that the  $f_i(N)$  for  $1 \leq i \leq n$  arising in the inhomogeneous parts of the recurrences (2.6) are given as a linear combination of the  $b_i(N)$ . Since they are given in the form (2.7) where the  $b_{i,j}$  are nested sum expressions over hypergeometric products, one can collect terms in  $\varepsilon$  and gets the  $\varepsilon$ -expansion

$$f_i(N) = f_{i,o}(N)\varepsilon^o + f_{i,o+1}(N)\varepsilon^{o+1} + \dots f_{i,o}(N)\varepsilon^{o+l_i} + \dots, \quad (2.9)$$

where the  $f_{i,j}(N)$  are expressions in terms of nested sums over hypergeometric products (which are free of  $\varepsilon$ ).

- (b) Given the initial values (2.8) and the recurrences (2.6) with (2.9) where the  $f_{i,j}(N)$  are given in terms of nested sums over hypergeometric products, we can activate our recurrence solver [20] based on [18, 19]. For all  $i$  with  $1 \leq i \leq n$ , we can constructively decide if the integral  $I_i(N)$  can be expressed in terms of nested product-sum expressions. If such a representation does not exist for some  $i$ , we stop and learned that our problem has to be formulated outside of the class of nested sums over hypergeometric products.

**Remark 1.** *The solutions in terms of nested sums over hypergeometric products are also called d'Alembertian solutions [18] which can be computed by using algorithms from [18, 19]. Further we remark that the sum representations produced by these algorithms are rather complicated: they are highly nested and the summands have denominators which do not factor nicely. Here we utilize symbolic summation algorithms [21] based on difference field theory [22] and difference ring theory [23] in order to rewrite the found solutions in terms of special functions like cyclotomic generalized harmonic sums or nested binomial sums.*

- (c) We can combine the computed  $\varepsilon$ -expansions of the  $I_1(N), \dots, I_n(N)$  and their shifted versions yielding the  $\varepsilon$ -expansions of the remaining integrals  $I_{n+1}(N), \dots, I_m(N)$  whose coefficients are given in terms of nested sums over hypergeometric products.

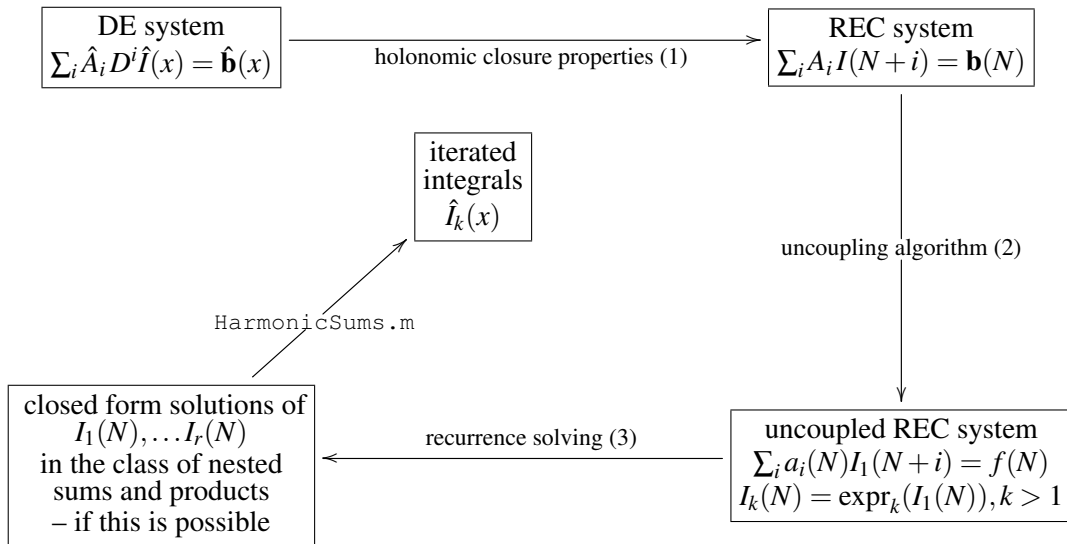
**Remark 2.** *Within intermediate steps but also for the final result it is important to get compact representations of the found nested sums and products. In general, we can exploit the underlying*

<sup>4</sup>Sometimes we need more initial values; the necessary number can be detected during the recurrence solving.

summation theory [23] in order to compute expressions in terms of nested sums and products where no algebraic (i.e., polynomial) relations exist among the occurring sums and products. Restricting to harmonic sums and cyclotomic sums, one can employ an alternative and very efficient machinery. Namely, it has been shown in [35] that such an optimal representation can be obtained using the underlying quasi-shuffle relations [36]. Currently, further investigations along the lines of [35] are carried out to avoid the rather involved difference ring algorithms and to reduce the calculations to quasi-shuffle relations and further properties given by the occurring alphabets of the nested sums.

- (4) Optionally, we can rewrite the power series solutions in terms of nested integrals involving the extra parameter  $x$  using the package `HarmonicSums`.

The above tactic can be summarized by the following diagram.



## 2.2 Tactic 2: Uncoupling a system of differential equations

In the above approach we transformed the system of differential equations to a system of recurrences in step (1), uncoupled the system in step (2) and finally decided if the unknown integrals can be given in terms of nested sums over hypergeometric products in steps (3) and (4). We remark that in step (1) one obtains usually a coupled system of higher-order difference equations. Uncoupling this system directly, or as proposed in our strategy, bringing it first to a first-order system and uncoupling it afterwards, worked very well for many concrete calculations. However, recently we entered examples where the calculation time of the known uncoupling algorithms blew up dramatically. In contrast to that, the original system provided by IBP methods is usually rather small (and in particular first-order), and the uncoupling algorithms applied to this differential equation system behaves rather tame.

Therefore a different and very promising tactic is to reverse steps (1) and (2). More precisely, we can proceed as follows.

- (1) The uncoupling algorithms available in `OreSys` [24] work for any Ore algebra covering besides the difference case also the differential case. Within `SolveCoupledSystem` we use again Zürcher's algorithm. But this time we uncouple immediately the first-order system (2.1) of differential equations. In most instances, the matrix  $\hat{A}$  is already invertible. Otherwise, we carry out a similar preprocessing step as worked out in Step (2b) of Subsection 2.1. Summarizing, one obtains  $n$  ( $1 \leq n \leq r$ ) scalar differential equations of the form

$$a_{i,r_i}(x)D_x^{r_i}\hat{I}_i(x) + a_{i,r_i-1}(x)D_x^{r_i-1}\hat{I}_i(x) + \cdots + a_{i,0}(x)D_x^0\hat{I}_i(x) = \sum_{j=1}^r \sum_k d_{i,j,k}(x)D_x^k\hat{b}_j(x) \quad (2.10)$$

where the sums on the right hand side are finite and the  $a_{i,j}(x)$  and  $d_{i,j,k}(x)$  are from  $\mathbb{K}[x]$ . In addition, the remaining integrals  $\hat{I}_{n+1}(x), \dots, \hat{I}_m(x)$  are related to the  $\hat{I}_j(x)$  with  $1 \leq i \leq n$  and the  $\hat{b}_i(x)$  with  $1 \leq i \leq r$  as follows: for all  $i$  with  $n < i \leq m$  we get

$$\hat{I}_i(x) = \sum_{j=1}^n \sum_k \alpha_{i,j,k}(x)D_x^k\hat{I}_j(x) + \sum_{j=1}^r \sum_k \beta_{i,j,k}(x)D_x^k\hat{b}_j(x) \quad (2.11)$$

where the sums on the right are finite and where the  $\alpha_{i,j,k}(x)$  and  $\beta_{i,j,k}(x)$  are given from  $\mathbb{K}(x)$ .

- (2) Next, we transform the scalar equations (2.10) to recurrences of the form (2.6) with  $a_{i,j}(N) \in \mathbb{K}[N]$  using the same tactic as used in Step (1) of Subsection 2.1. Under the assumption that the  $b_i(N)$  have an  $\varepsilon$ -expansion (2.7) where the first coefficients  $b_{i,j}(N)$  can be given in terms nested sums over hypergeometric products, we can derive  $\varepsilon$ -expansions (2.9) of the inhomogeneous parts of the recurrences (2.6).
- (3) Afterwards we apply Step (3) of Subsection 2.1. Namely, together with the initial values (2.8) with  $v = 1, \dots, m_i$  we can decide algorithmically if the coefficients  $I_{i,j}(N)$  of (2.8) for  $1 \leq i \leq n$  can be represented in terms of nested sums over hypergeometric products.
- (4) Finally, we compute the coefficients  $I_i(N)$  of the integrals (1.1) for  $n < i \leq m$  as follows. We replace the  $\hat{I}_i(x)$  and  $\hat{b}_i(x)$  on the right hand side of (2.11) by their power series representations (1.1) and (2.2). In addition, we replace the  $I_i(N)$  and  $b_i(N)$  arising in the power series representations further by the given representation in terms of nested sums over hypergeometric products (possibly in its  $\varepsilon$ -expansion). Then we utilize the package `SumProduction` [21] that contains rather efficient algorithms [7] to calculate the  $N$ th coefficient of the corresponding power series representation. First, we separate the expressions accordingly and get the representation

$$\hat{I}_i(x) = \varepsilon^\omega \hat{I}_{i,\omega}(x) + \varepsilon^{\omega+1} \hat{I}_{i,\omega+1}(x) + \cdots + \varepsilon^{\omega+l_i} \hat{I}_{i,\omega+l_i}(x) \quad (2.12)$$

for some  $\omega \in \mathbb{Z}$  where each  $\hat{I}_{i,j}(x)$  can be written as a finite sum consisting of summands of the form

$$\hat{H}(x) = \hat{q}(x) \sum_{N=0}^{\infty} x^N h_1(N) \dots h_l(N). \quad (2.13)$$

Here we have  $\hat{q}(x) \in \mathbb{K}'(x)$  and for all  $1 \leq i \leq l$  we have that  $h_i(N) \in \mathbb{K}'(x)$ ,  $h_i(N)$  is a hypergeometric product or  $h_i(N)$  is a nested sum over hypergeometric products. With a brute force approach one can now compute the  $N$ th coefficient for each such expression as given in (2.13),



i.e., we can compute an expression  $H(N)$  in terms of nested sums over hypergeometric products such that

$$\hat{H}(x) = \sum_{N=0}^{\infty} H(N)x^N$$

holds. This finally yields the  $N$ th coefficient  $I_{i,j}(N)$  of

$$\hat{I}_{i,j}(x) = \sum_{N=0}^{\infty} I_{i,j}(N)x^N$$

in terms of nested sums over hypergeometric products.

Namely, first we expand  $q(x) = \frac{a(x)}{b(x)}$  with  $a(x), b(x) \in \mathbb{K}'[x]$  in a series expansion

$$\hat{q}(x) = \sum_{N=\mu}^{\infty} q(N)x^N \quad (2.14)$$

with  $\mu \in \mathbb{Z}$  as follows [7]. Consider the complete factorization

$$b(x) = cx^{v_0}(x - \rho_1)^{v_1}(x - \rho_2)^{v_2} \dots (x - \rho_v)^{v_v} \quad (2.15)$$

with  $c \in \mathbb{K}' \setminus \{0\}$  and  $\rho_i \in \bar{\mathbb{K}} \setminus \{0\}$  ( $\bar{\mathbb{K}}$  is the algebraic closure of  $\mathbb{K}'$ ) where  $v_i \in \mathbb{N}$  counts the multiplicity of the roots  $\rho_i$  ( $\rho_0 = 0$ ). Then, as worked out in [37, Thm. 4 1.1], we can calculate the expansion

$$\frac{1}{b(x)} = \frac{1}{x^{v_0}} \sum_{N=0}^{\infty} \beta(N)x^N$$

with

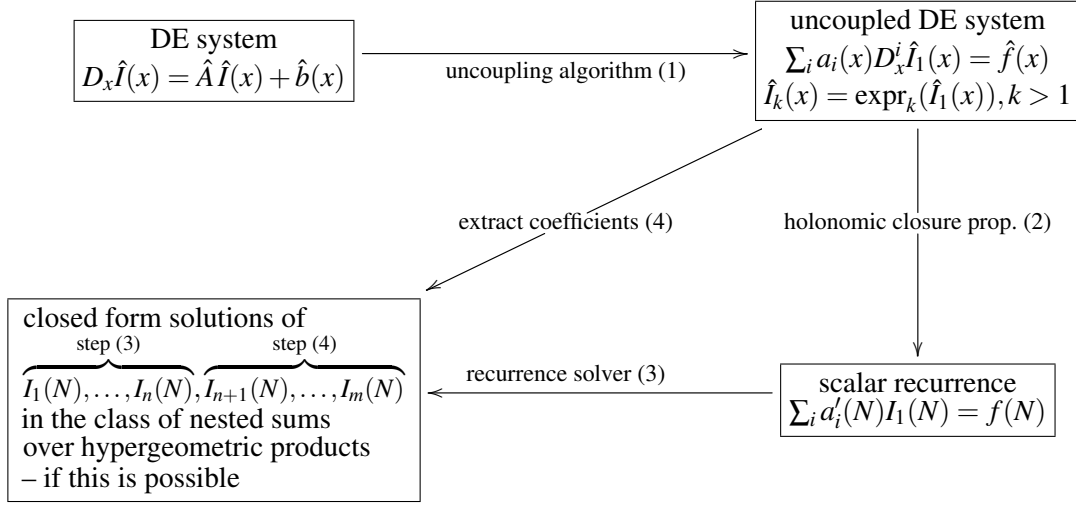
$$\beta(N) = p_1(N)\rho_1^N + p_2(N)\rho_2^N + \dots + p_v(N)\rho_v^N,$$

where the  $p_i(N)$  are polynomials in  $N$  with degree at most  $v_i - 1$ . Now we perform the Cauchy product on  $\hat{q}(x) = a(x) \frac{1}{x^{v_0}} \sum_{N=0}^{\infty} \beta(N)x^N$ , and it follows that the coefficient  $q(N)$  of the expansion (2.14) can be written again as a linear combination of the  $\rho_i^N$  with polynomial coefficients in  $N$ . Finally, we obtain

$$H(N) = \sum_{k=\mu}^{N+v_0} h_1(k)h_2(k) \dots h_l(k)r(k)q(N-k)$$

by applying once more the Cauchy product. Since  $q(N-k)$  is given as a linear combination of the  $\rho_i^{N-k} = \rho_i^N \rho_i^{-k}$  where the coefficients are polynomials in  $N$  and  $k$ , we can pull out all expressions that depend on  $N$ . Summarizing, we can write  $H(N)$  as an expression in terms of nested sums over hypergeometric products. In particular, the summands of the arising sums are built by the objects  $h_1(k)h_2(k) \dots h_l(k)$  given in (2.12,2.13) and the roots  $\rho_i^k$  from (2.15).

In short, we can summarize the second approach with the following diagram.



### 2.2.1 Improvement 1: compute the $N$ th coefficients efficiently (step (4))

Tactic 2 has the advantage that one uncouples the system of differential equations that one is given, e.g., by IBP methods. They are usually in a rather nice shape and can be uncoupled in many examples extremely efficiently. However, this big advantage is paid by the challenge to compute the  $N$ th coefficient of the  $\hat{I}_{i,j}(x)$  in (2.12): IBP methods usually calculate expressions with rather complicated denominators, i.e., one obtains subexpressions  $\hat{H}(x)$  in (2.13) where the rational functions  $\hat{q}(x) = \frac{a(x)}{b(x)}$  have denominators  $b(x)$  that do not have nice irreducible factors over  $\mathbb{K}'$  (mostly over  $\mathbb{Q}$ ). As a consequence, the method proposed in step (4) yields alien sums whose summands are not expected to appear within the final result.

*Example.* Consider

$$\hat{H}(x) = \frac{1}{1-ax} \sum_{N=0}^{\infty} x^N S_{2,1}(N) = \sum_{N=0}^{\infty} H(N) x^N$$

with  $a \in \mathbb{Z}$ . Then the  $N$ th coefficient is

$$H(N) = \begin{cases} \frac{a^{N+1}}{a-1} S_{2,1}\left(\frac{1}{a}, 1, N\right) - \frac{1}{a-1} S_{2,1}(N) & \text{if } a \in \mathbb{Z} \setminus \{0, 1\} \\ S_{2,1}(N) & \text{if } a = 0 \\ (N+1)S_{2,1}(N) - \frac{1}{2}S_2(N) - \frac{1}{2}S_1(N)^2 & \text{if } a = 1. \end{cases}$$

The arising sums with  $a = 0, \pm 1$  pop up almost everywhere within our calculations [25]. So denominators of the form  $1-ax$  with  $a = 0, \pm 1$  are no surprise. In some instances, also the generalized sums with  $a = \pm 2$  arise, and thus also the underlying denominators will appear. However, sums coming from  $a \in \mathbb{Z}$  and  $a \neq 0, \pm 1, \pm 2$  have not occurred in our ongoing calculations, but will arise within intermediate calculations when one executes step (4). The situation gets worse if non-linear factors are treated. One of the simplest cases is  $\hat{q}(x) = \frac{1}{1-x-x^2}$ . In this situation, e.g., the subexpression

$$\frac{1}{1-x-x^2} \sum_{N=0}^{\infty} x^N S_{2,1}(N) = \sum_{N=0}^{\infty} H(N) x^N$$

has the  $N$ th coefficient

$$H(N) = \frac{3\sqrt{5}-5}{10}(-1)^N \Phi^{-N-1} S_{2,1}(-\Phi, 1, N) + \frac{5+3\sqrt{5}}{10} \phi^{-N-1} S_{2,1}(\phi, 1, N) - S_{2,1}(N) \quad (2.16)$$

with the golden ration  $\Phi = \frac{\sqrt{5}+1}{2}$  and  $\phi = \frac{\sqrt{5}-1}{2}$ .

One option is to produce the final expression with all these artificial sums. Then eliminating all relations among these sums (see Remark 2) will lead to an expression where all alien sums will collapse and the expected sums will remain. However, in many calculations the number of sums in the summands (2.13) can be quite large – we considered cases with up to 1000 sums. Performing than step (4) naively the number of sums will explode. Even worse, if algebraic numbers like  $\phi$  and  $\Phi$  arise, the computation of algebraic relations turns into a real computer algebra challenge. In order to avoid these troubles, the following alternative option worked out in [7] has been incorporated into the package `SumProduction` that we use heavily as a subroutine within our newly developed package `SolveCoupledSystem`. Recall that the desired coefficients  $\hat{I}_{i,j}(x)$  of (2.12) can be written as a big expressions summed up by subexpressions of the form (2.13). We truncate now the infinite sums within the given expression of  $\hat{I}_{i,j}(x)$ , i.e., the arising sums are of the form

$$\sum_{N=0}^A x^N h_1(N) \dots h_l(N)$$

(instead of the form (2.13)). Now we eliminate all algebraic relations among these nested sums over hypergeometric products (involving in addition the parameter  $x$ ). Here the following magic happens: in all our examples, we observed that all sums that would contribute to alien terms as given in (2.16) vanish.

Still, the proposed tactic is rather expensive to treat all truncated sums (all with the same upper bound  $A$ ) simultaneously. As a compromise we filter out only those subexpressions that might contribute to alien sums, more precisely, we partition the subexpressions into several parts: (a) sums with denominators that have only nice irreducible factors (over  $\mathbb{Q}$ ) in the denominator, which we keep untouched (this is usually the largest part of the full expression), and (b) sums with denominators that have only bad irreducible factors in the denominator, i.e., which are of the form  $(1-ax)$  with  $a \neq 0, \pm 1$  (or even  $a \neq \pm 2$ ) and factors which are not linear (over  $\mathbb{Q}$ ). Even more, we partition the sums with bad denominators further such that sums are collected which have common bad factors. Then we compute for each such partition of bad sums, which consists of conquerable subexpressions, all algebraic relations. In all our examples all these bad sums within their clusters vanish. Finally, we compute the limit  $A \rightarrow \infty$  and end up at an alternative expression of  $\hat{I}_{i,j}(x)$  where now all the unwanted sums are gone. Computing finally the  $N$ th coefficient as worked out in step (4) will lead to an expression where the nested sums over hypergeometric products have nice denominators. In particular, the number of these sums is now manageable, and we can compute an alternative representation in terms of nested sums where all arising sums are algebraically independent among each other (see Remark 2).

## 2.2.2 Improvement 2: compute recurrence relations of smaller order (step 2))

Recall the first steps of our proposed procedure. Suppose we obtained the scalar differential equation (2.10) in step (1) with  $a_{i,j}(x)$  and  $d_{i,j,k}(x)$  being from  $\mathbb{K}[x]$ . Then plugging in (1.1) into (2.10),

using the rule (2.3) and taking the  $N$ th coefficient, we end up at a linear recurrence of the form (2.4) where the  $f_i(N)$  are given as a linear combination of the  $b_1(N), \dots, b_r(N)$  over  $\mathbb{K}(N)$  with possible shifts in  $N$ . Under the assumption that the  $b_i(N)$  have an  $\varepsilon$ -expansion (2.7) where the first coefficients  $b_{i,j}(N)$  can be given in terms nested sums over hypergeometric products, we can derive an  $\varepsilon$ -expansion (2.9) of the inhomogeneous part of the recurrence (2.6).

Observe that the order of the derived recurrence is bounded by the maximum of the degrees of the coefficients  $a_{i,j}(x)$ . If the order is not too large, the method proposed above works perfectly fine. However, in recent examples we calculated recurrences of order 16 or higher, and it is then almost impossible to calculate 16 (or more) initial values that are needed for step (3). Luckily, in all these examples it turns out that  $d(x) = \gcd(a_{i,0}(x), \dots, \gcd(a_{i,r_i}(x))) \in \mathbb{K}[x]$  has a rather high degree. Hence dividing (2.10) through  $d(x)$  leads to the differential equation

$$a'_{i,r_i}(x)D_x^{r_i}\hat{I}_i(x) + a'_{i,r_i-1}(x)D_x^{r_i-1}\hat{I}_i(x) + \dots + a'_{i,0}(x)D_x^0\hat{I}_i(x) = \sum_{j=1}^r \sum_k d'_{i,j,k}(x)D_x^k \hat{b}_j(x) \quad (2.17)$$

where the  $a'_{i,j} = \frac{a_{i,j}(x)}{d(x)} \in \mathbb{K}[x]$  have substantially smaller degrees and where  $d'_{i,j,k} = \frac{d_{i,j,k}}{d(x)} \in \mathbb{K}(x)$ . Hence taking the  $N$ th coefficient on the left hand side of (2.17) will yield a difference operator in  $I_i(N)$  whose order is substantially smaller (the order is bounded by the maximum of the degrees of the coefficients  $a'_{i,j}(x)$ ). However, in order to get the  $N$ th coefficient on the right hand side of (2.17), i.e., in order to get a recurrence of the form (2.6), further calculations are necessary. Since  $d'_{i,j,k} \in \mathbb{K}(x)$  is usually not a polynomial in  $x$ , we have to apply again the rather involved calculations steps as sketched in step (4). In this regard, the improvements of Section 2.2.1 play a central role to carry out these calculations efficiently and to provide a linear recurrence with a substantially smaller recurrence order for step (2) of our procedure.

### 3. Conclusion

In both algorithms, presented in Subsections 2.1 and 2.2 respectively, one ends up at scalar recurrences (2.6): sometimes they are the same, sometimes one or the other method finds a better recurrence (with smaller coefficient size or with lower recurrence order). This suggests to apply both tactics (up to a certain point) and to execute the version in full detail that is more appropriate for the concrete problem. As mentioned already above, a central advantage of the second tactic is that one can uncouple the system straightforwardly (without any preprocessing steps as sketched in Subsection 2.1 that might blow up the system). In some instances this leads to a much better space–time behavior. However, in this approach one has to compute the  $N$ th coefficient of the remaining integrals, which again can be rather time consuming. But using our sophisticated symbolic summation technologies (see Subsection 2.2.1) this problem turns out to be feasible in many examples. The second advantage of our new method is that one might find recurrences with smaller orders (see Subsection 2.2.2). As a consequence, one needs less initial values to determine the respective master integrals. Since the calculations of such initial values is rather challenging, we expect that this last feature will support future calculations.

### References

- [1] A. von Manteuffel and C. Studerus, arXiv:1201.4330 [hep-ph];

- C. Studerus, *Comput. Phys. Commun.* **181** (2010) 1293 [arXiv:0912.2546 [physics.comp-ph]].
- [2] S. Laporta, *Int. J. Mod. Phys. A* **15** (2000) 5087 [hep-ph/0102033].
- [3] K.G. Chetyrkin, A.L. Kataev and F.V. Tkachov, *Nucl. Phys. B* **174** (1980) 345.
- [4] A.V. Kotikov, *Phys. Lett. B* **254** (1991) 158;  
M. Caffo, H. Czyz, S. Laporta and E. Remiddi, *Acta Phys. Polon. B* **29** (1998) 2627 [hep-th/9807119]; *Nuovo Cim. A* **111** (1998) 365 [hep-th/9805118];  
T. Gehrmann and E. Remiddi, *Nucl. Phys. B* **580** (2000) 485 [hep-ph/9912329].
- [5] A.V. Kotikov, In : *Subtleties in quantum field theory*, ed. D. Diakonov, 150–174 [arXiv:1005.5029 [hep-th]]; *Theor. Math. Phys.* **176** (2013) 913; [arXiv:1212.3732 [hep-ph]]. *Phys. Part. Nucl.* **44** (2013) 374;  
J.M. Henn, *Phys. Rev. Lett.* **110** (2013) 251601 [arXiv:1304.1806 [hep-th]].
- [6] C. Schneider, A. De Freitas and J. Blümlein, *PoS LL* **2014** (2014) 017 [arXiv:1407.2537 [cs.SC]].
- [7] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel and C. Schneider, *Comput. Phys. Commun.* **202** (2016) 33 [arXiv:1509.08324 [hep-ph]].
- [8] J. Ablinger, J. Blümlein, A. de Freitas and C. Schneider, In: *Proc. of 12th International Symposium on Radiative Corrections (Radcor 2015) PoS(RADCOR2015)060*, pp. 1-13 (2015), arXiv:1601.01856 [cs.SC].
- [9] J. Blümlein and S. Kurth, *Phys. Rev. D* **60** (1999) 014018 [arXiv:hep-ph/9810241].  
J.A.M. Vermaseren, *Int. J. Mod. Phys. A* **14** (1999) 2037 [arXiv:hep-ph/9806280].
- [10] S. Moch, P. Uwer and S. Weinzierl, *J. Math. Phys.* **43** (2002) 3363 [hep-ph/0110083].
- [11] J. Ablinger, J. Blümlein and C. Schneider, *J. Math. Phys.* **54** (2013) 082301 [arXiv:1302.0378 [math-ph]].
- [12] J. Ablinger, J. Blümlein and C. Schneider, *J. Math. Phys.* **52** (2011) 102301 [arXiv:1105.6063 [math-ph]].
- [13] J. Fleischer, A.V. Kotikov and O.L. Veretin, *Nucl. Phys. B* **547** (1999) 343 [hep-ph/9808242];  
A. I. Davydychev and M. Y. Kalmykov, *Nucl. Phys. B* **699** (2004) 3 [arXiv:0303162 [hep-th]];  
S. Weinzierl, *J. Math. Phys.* **45** (2004) 2656 [arXiv:0402131 [hep-ph]].
- [14] J. Ablinger, J. Blümlein, C.G. Raab and C. Schneider, *J. Math. Phys.* **55** (2014) 112301 [arXiv:1407.1822 [hep-th]].
- [15] J. Ablinger, J. Blümlein and C. Schneider, *J. Phys. Conf. Ser.* **523** (2014) 012060 [arXiv:1310.5645 [math-ph]];  
J. Ablinger and J. Blümlein, in: *Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions*, C. Schneider, J. Blümlein, Eds., p. 1, (Springer, Wien, 2013) arXiv:1304.7071 [math-ph].
- [16] B. Zürcher, *Rationale Normalformen von pseudo-linearen Abbildungen*, Master's thesis, Mathematik, ETH Zürich (1994).
- [17] A. Danilevskiĭ, *Mat. Sbornik* **2** (1937) 169;  
M.A. Barkatou, *Appl. Algebra Engrg. Comm. Comput.* **4**(3) (1993) 185;  
M. Bronstein and M. Petkovšek, *Theor. Comput. Sci.*, **157** (1) (1996) 3;  
S.A. Abramov and E.V. Zima, *Proc. Int. Conf. on Computational Modelling and Computing in Physics*, Dubna, RU, Sept. 16-26 (1996) 16;  
A. Bostan, F. Chyzak, E. de Panafieu, *ISSAC'13 June 26-29, 2013, Boston*, arXiv:1301.5414 [cs.SC] and references therein.
- [18] M. Petkovšek, *J. Symbolic Comput.* **14** (1992) 243;  
S.A. Abramov and M. Petkovšek, in: J. von zur Gathen (ed.) *Proc. ISSAC'94*, 169 ACM Press (1994) pp. 169.  
P.A. Hendriks and M.F. Singer, *J. Symbolic Comput.*, **27**(3) (1999) 239.
- [19] M. Bronstein, *J. Symbolic Comput.*, **29**(6) (2000) 841. C. Schneider, *Symbolic Summation in Difference Fields* Ph.D. Thesis RISC, Johannes Kepler University, Linz technical report 01-17 (2001).  
C. Schneider, *An. Univ. Timisoara Ser. Mat.-Inform.* **42** (2004) 163;  
C. Schneider, *J. Differ. Equations Appl.* **11** (2005) 799;  
C. Schneider *Appl. Algebra Engrg. Comm. Comput.* **16**(2005) 1.

- [20] J. Blümlein, S. Klein, C. Schneider and F. Stan, *J. Symbolic Comput.* **47** (2012) 1267 [arXiv:1011.2656 [cs.SC]];
- [21] C. Schneider, *Sém. Lothar. Combin.* **56** (2007) 1, article B56b;  
C. Schneider, *J. Phys. Conf. Ser.* **523** (2014) 012037 [arXiv:1310.0160 [cs.SC]];  
C. Schneider, Simplifying Multiple Sums in Difference Fields, in: *Computer Algebra in Quantum Field Theory: Integration, Summation and Special Functions Texts and Monographs in Symbolic Computation* eds.  
C. Schneider and J. Blümlein (Springer, Wien, 2013) 325 [arXiv:1304.4134 [cs.SC]];
- [22] M. Karr, *J. ACM* **28** (1981) 305;  
C. Schneider, *J. Symbolic Comput.* **43** (2008) 611, [arXiv:0808.2543v1];  
C. Schneider, in: *Motives, Quantum Field Theory, and Pseudodifferential Operators, Clay Mathematics Proceedings Vol. 12* ed. A. Carey, D. Ellwood, S. Paycha and S. Rosenberg, (Amer. Math. Soc) (2010), 285 [arXiv:0904.2323];  
C. Schneider, in: *Computer Algebra and Polynomials, Applications of Algebra and Number Theory*, J. Gutierrez, J. Schicho, M. Weimann (ed.), *Lecture Notes in Computer Science (LNCS)* 8942 (2015), 157 [arXiv:13077887 [cs.SC]];  
C. Schneider, *Ann. Comb.* **14** (2010) 533 [arXiv:0808.2596].
- [23] M. van der Put and M. F. Singer. *Galois theory of difference equations*, volume 1666 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1997;  
C. Schneider, in: *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014, 15th International Symposium*, F. Winkler, V. Negru, T. Ida, T. Jebelean, D. Petcu, S. Watt, D. Zaharie (ed.), (2015) pp. 26; IEEE Computer Society, arXiv:1412.2782v1 [cs.SC];  
C. Schneider, *J. Symb. Comput.* **72** (2016) 82, doi:10.1016/j.jsc.2015.02.002 [arXiv:1408.2776 [cs.SC]];  
Summation Theory II: Characterizations of  $R$ ???-extensions and algorithmic aspects *J. Symb. Comput.*, in press, 2016. doi:10.1016/j.jsc.2016.07.028 [arXiv:1603.04285 [cs.SC]].
- [24] S. Gerhold, Uncoupling systems of linear Ore operator equations, Master's thesis, RISC, J. Kepler University, Linz, 2002.
- [25] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel and C. Schneider, *Nucl. Phys. B* **890** (2014) 48 [arXiv:1409.1135 [hep-ph]];  
J. Ablinger, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel, M. Round, C. Schneider and F. Wißbrock, *Nucl. Phys. B* **882** (2014) 263 [arXiv:1402.0359 [hep-ph]];  
J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel, M. Round, C. Schneider and F. Wißbrock, *Nucl. Phys. B* **886** (2014) 733 [arXiv:1406.4654 [hep-ph]];  
A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel and C. Schneider, *Nucl. Phys. B* **897** (2015) 612 [arXiv:1504.08217 [hep-ph]];  
A. Behring, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel and C. Schneider, *Phys. Rev. D* **92** (2015) 11, 114005 [arXiv:1508.01449 [hep-ph]];  
J. Ablinger, J. Blümlein, S. Klein, C. Schneider and F. Wißbrock, *Nucl. Phys. B* **844** (2011) 26 [arXiv:1008.3347 [hep-ph]];  
A. Behring, I. Bierenbaum, J. Blümlein, A. De Freitas, S. Klein and F. Wißbrock, *Eur. Phys. J. C* **74** (2014) no.9, 3033 [arXiv:1403.6356 [hep-ph]];  
A. Behring, J. Blümlein, G. Falcioni, A. De Freitas, A. von Manteufel, and C. Schneider, DESY 16–148.
- [26] J. Blümlein, M. Kauers, S. Klein and C. Schneider, *Comput. Phys. Commun.* **180** (2009) 2143 [arXiv:0902.4091 [hep-ph]].
- [27] J. Ablinger, J. Blümlein, A. De Freitas, A. Hasselhuhn, A. von Manteuffel, M. Round and C. Schneider, *Nucl. Phys. B* **885** (2014) 280 [arXiv:1405.4259 [hep-ph]].
- [28] J. Blümlein and J.A.M. Vermaseren, *Phys. Lett. B* **606** (2005) 130 doi:10.1016/j.physletb.2004.11.059 [hep-ph/0411111];  
B. Ruijl, T. Ueda, J.A.M. Vermaseren, J. Davies and A. Vogt, arXiv:1605.08408 [hep-ph].
- [29] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, M. van Hoeij, C.G. Raab, and C. Schneider, DESY 16–147.

- [30] J. Ablinger, A Computer Algebra Toolbox for Harmonic Sums Related to Particle Physics, Diploma Thesis, J. Kepler University Linz, 2009, arXiv:1011.1176 [math-ph];  
J. Ablinger, PoS LL **2014** (2014) 019.  
J. Ablinger, arXiv:1606.02845 [cs.SC], to appear in Proc. Loops and Legs, 2016.
- [31] M. Apagodu and D. Zeilberger, Adv. Appl. Math. (Special Regev Issue), **37** (2006) 139;  
J. Ablinger, Computer Algebra Algorithms for Special Functions in Particle Physics, Ph.D. Thesis, J. Kepler University Linz, 2012, arXiv:1305.0687 [math-ph].
- [32] S.A. Abramov and M. Barkatou, Proceedings of ISSAC'98 (Rostock), 1998;  
S.A. Abramov and E. Khmelnov, Programming and Computer Software **38** (2012), no. 2, 84–91.
- [33] J. Middeke, C. Schneider, To appear in ACM Communications in Computer Algebra, 2016. Extended abstract of the poster presentation at ISSAC 2016.
- [34] S. Weinzierl, 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 [arXiv:13016918 [hep-ph]].
- [35] J. Ablinger, C. Schneider. Algebraic independence of (cyclotomic) harmonic sums, 2015. arXiv:1510.03692 [cs.SC].
- [36] J. Blümlein, Comput. Phys. Commun. **159** (2004) 19 [hep-ph/0311046].
- [37] R. Stanley, Enumerative Combinatorics, Vol. **1**, (Cambridge University, Press, 1997).