

Challenges for analytic calculations of the massive three-loop form factors

J. Blümlein,^a A. De Freitas,^{b,*} P. Marquard^a and C. Schneider^b

^a*Deutsches Elektronen-Synchrotron DESY, Platanenallee 6, 15738 Zeuthen, Germany*

^b*Johannes Kepler University Linz, Research Institute for Symbolic Computation (RISC),
Altenbergerstraße 69, A-4040, Linz, Austria*

*E-mail: johannes.bluemlein@desy.de, afreitas@risc.jku.at,
peter.marquard@desy.de, carsten.schneider@risc.jku.at*

The calculation of massive three-loop QCD form factors using in particular the large moments method has been successfully applied to quarkonic contributions in [1]. We give a brief review of the different steps of the calculation and report on improvements of our methods that enabled us to push forward the calculations of the gluonic contributions to the form factors.

ARXIV HEP-PH: [arXiv:2408.07046](https://arxiv.org/abs/2408.07046)

DESY-24-121

RISC REPORT NUMBER 24-05

PoS (LL2024) 031

Loops and Legs in Quantum Field Theory (LL2024)

14-19, April, 2024

Wittenberg, Germany

*Speaker

1. Introduction

Massive QCD form factors constitute basic building blocks for several important processes in particle physics, such as heavy quark production at e^+e^- colliders and Higgs or gauge boson decay into heavy quarks. In order to match the precision of available experimental results, these form factors must be computed at higher orders in the corresponding perturbative expansions. The two-loop contributions were computed up to $O(\varepsilon^0)$ about two decades ago in Refs. [2–5], where ε is the dimensional regularization parameter, which is given by $\varepsilon = (4 - D)/2$, where D is the dimension. This calculation was later extended to $O(\varepsilon)$ in [6] and to $O(\varepsilon^2)$ in [7]. These higher orders in ε are needed in order to combine them with the corresponding higher loop order calculations. The first analytic results at three loops were obtained in the planar limit in Refs. [8–10]. The high energy behaviour of three loop form factors was studied in Refs. [11, 12], and light-fermion contributions were computed in Refs. [13, 14]. Analytic results in terms of harmonic polylogarithms [15] for the solvable parts of quarkonic contributions (that is, contributions from diagrams containing a closed fermion loop) were given in [16]. More recently, numerical results for all form factors, including singlet and anomaly contributions, were obtained in Refs. [17–19]. Later, in Ref. [1], we used the large moments method [20–22] to obtain analytic results for the quarkonic contributions to all non-singlet form factors. In these proceedings, we report on the progress using this method to compute the gluonic contributions, that is, the contributions stemming from diagrams with no internal closed fermion loops.

The large moments method derives its name from the fact that it was first applied to the computation of the factorizable parts of massive three-loop operator matrix elements and related problems using a large number of Mellin moments [23–28]. This method can also be applied if we have a large number of expansion coefficients, instead of Mellin moments, for a physical quantity depending on a single variable. We will give a brief review of this method in Section 2 and focus on concrete challenges of the large moments method related to the form factor project in Section 3. In Section 4 we will present the progress we have made applying this method to the gluonic case, which is considerably more involved than the quarkonic one. In Section 5, we give the conclusions.

2. Review of the method

The three-point functions with an external massive quark-anti-quark pair coupling to a vector, axial-vector, scalar or pseudo-scalar boson are given (in that order) by

$$\Gamma_V^\mu = \gamma^\mu F_{V,1}(\hat{s}) - \frac{i}{2m} \sigma^{\mu\nu} q_\nu F_{V,2}(\hat{s}) , \quad (1)$$

$$\Gamma_A^\mu = \gamma^\mu \gamma_5 F_{A,1}(\hat{s}) - \frac{1}{2m} q^\mu \gamma_5 F_{A,2}(\hat{s}) , \quad (2)$$

$$\Gamma_S = m F_S(\hat{s}) , \quad (3)$$

$$\Gamma_P = im F_P(\hat{s}) ; \quad (4)$$

here $\hat{s} = q^2/m^2$, where q is the momentum of the boson and m is the heavy quark mass. The functions $F_{V,1}$, $F_{V,2}$, $F_{A,1}$, $F_{A,2}$, F_S and F_P are the so called form factors, which can be extracted from Eqs. (1-4) by applying suitable projectors. The first steps of our calculation are standard:

We generated the Feynman diagrams using QGRAF [29], and after introducing the aforementioned projectors, we performed the Lorentz algebra using FORM [30, 31] and the color algebra using Color [32]. After this, we end up with expressions for the form factors in terms of linear combinations of scalar Feynman integrals, which we reduced to master integrals using Crusher [33]. At this point, a system of first order differential equations obeyed by the master integrals can be obtained. In many similar perturbative calculations, the corresponding systems of differential equations can be solved in terms of closed form solutions involving harmonic polylogarithms and their generalizations [15, 34–41]. Unfortunately, this is not always possible, which is often the case in calculations involving massive particles, such as the present one. If we try to decouple the system using, e.g., the package OreSys [42], we find out that the resulting higher order differential equations are not first order factorizable [43], leading to complicated solutions involving elliptic or even higher transcendental functions, as well as iterated integrals over these functions, which can be difficult to use for numerical evaluations and phenomenological purposes. For this reason, one is forced to adopt a different tactic. In Refs. [17–19], the authors tackled the problem by using the differential equations satisfied by the master integrals in order to find approximate numerical solutions to the master integrals consisting of power and power-log expansions with numerical coefficients obtained by matching the solutions at different values of \hat{s} . In Ref. [1], we used a strategy also based on differential equations, but instead of using the differential equations obeyed by the master integrals. Using guessing algorithms [44], we derived recursion relations and differential equations from the sequences of rational numbers multiplying the constants appearing in the expansion at $x = 1$ of the form factors, where,

$$\hat{s} = -\frac{(1-x)^2}{x}. \quad (5)$$

The form factors can be written as follows

$$F_I(x) = \sum_{k,l} C_k \mathcal{K}_l F_{I;k,l}(x), \quad (6)$$

where I labels any of the possible form factors, the \mathcal{K}_l 's are any of the constants in the following list

$$\mathcal{K}_l \in \left\{ 1, \zeta_2, \zeta_3, \ln(2)\zeta_2, \zeta_2^2, \ln^2(2)\zeta_2, \text{Li}_4\left(\frac{1}{2}\right), \ln^4(2), \zeta_2\zeta_3, \zeta_5 \right\}, \quad (7)$$

where the $\zeta_k = \zeta(k)$, i.e., the Riemann ζ function evaluated at integer values, and the C_k 's are color factors, which at three loops can take any of the following values

$$C_k \in \left\{ C_A^2 C_F, C_A C_F^2, C_F^3, n_l T_F C_F^2, n_l T_F C_F C_A, n_l^2 T_F^2 C_F, n_l n_h T_F^2 C_F, n_h^2 T_F^2 C_F, n_h T_F C_F^2, n_h T_F C_F C_A \right\}, \quad (8)$$

with

$$C_A = N_c, \quad C_F = \frac{N_c^2 - 1}{2N_c}, \quad T_F = \frac{1}{2}, \quad (9)$$

where N_c is the number of colors, while n_l and n_h are the number of light and heavy quarks, respectively. The first three color factors in (8) are the ones appearing in the gluonic case, while the remaining ones correspond to the quarkonic case.

The split in terms of the constants in Eq. (6), is done in such a way that the functions $F_{I;k,l}(x)$ can be expanded around $x = 1$ (which corresponds to $\hat{s} = 0$) as follows

$$F_{I;k,l}(x) = \sum_{i=0}^{\infty} r_{I;i}^{(k,l)}(0) y^i, \quad (10)$$

where $y = 1 - x$, and the $r_{I;i}^{(k,l)}(0)$ are rational numbers. Alternatively, we may also work with expansions around $\hat{s} = 0$, in which case, we will have equations (6) and (10) with x and y replaced by \hat{s} , and with different sequences of rational numbers in (10). In the quarkonic case, and more specifically, in the case of the last two color factors in (8), we were able to generate up to 8000 expansion coefficients by using the large moments method; for details we refer to Section 3. Using this information, we follow the guess and solve strategy that has been exemplified for the first time rigorously in particle physics in [28]. More precisely, using the 8000 coefficients, we succeeded in guessing recursion relations as well as differential equations associated with these sequences of rational numbers using the Sage package `ore_algebra` [44]. These recursions and differential equations were solvable for all but three of the constants in (7), namely,

$$\{1, \zeta_2, \zeta_3\}. \quad (11)$$

In all other cases, we solved the recursions by using the summation package `Sigma` [45] containing the algorithms introduced in [46, 47] and the references therein. Finally, we performed the infinite sum for the corresponding constants using `Sigma` and the package `HarmonicSums` [15, 34–41]. In the case of the constants in (11), since we cannot solve the corresponding recursions, we proceeded as follows:

1. We used the recursion relations to generate a much larger number of expansion coefficients for the expansion at $x = 1$. This allows us to evaluate this expansion with very high precision even at values very close to $x = 0$.
2. An ansatz in terms of a power-log expansion at $x = 0$ was inserted in the differential equation for each constant in (11). This allowed us to express all coefficients of the expansion at $x = 0$ in terms of only a few of them (as many as the order of the differential equation).
3. The expansions at $x = 1$ and $x = 0$ were matched at a point close to zero, and the coefficients left from the previous step were evaluated numerically with very high precision, making it possible to determine all of them in terms of known constants using the PSLQ algorithm [48, 49].
4. The same procedure was repeated for the expansions at the 2-particle and the 4-particle thresholds, albeit in these cases we worked with the expansions at $\hat{s} = 0$, and the expansions at $\hat{s} = 4$ and $\hat{s} = 16$ were given in terms of the variables $\sqrt{4 - \hat{s}}$ and $\sqrt{16 - \hat{s}}$, respectively. In the case of the 2-particle threshold, we also used the PSLQ algorithm to determine the coefficients, but new constants needed to be introduced, which were chosen among a few of the coefficients themselves.

We followed these steps for all of the form factors, allowing us to cover the full kinematic range in \hat{s} . In Ref. [1], we even went a bit further, and after combining all the results for all the constants in (6), we repeated what we did with the sequences of rational numbers associated to the expansions at $x = 1$, but now with the resulting sequences of rational numbers in the power-log expansions at $x = 0$, and again obtained the corresponding recursions relations, several of which could be solved and summed with `Sigma`, leading to new analytic results at that expansion point.

3. Challenges arising in the large moments method

In the following we present the basic idea of the large moments method [20] and elaborate on improvements that supplement the ideas given in [1, 16, 21, 22]. Starting with a physical expression¹ $\hat{P}(x)$ in terms of Feynman integrals, the IBP approach utilizes integration by parts (IBP) methods [33, 50, 51] and returns an alternative representation

$$\hat{P}(x) = q_1(x)\hat{I}_1(x) + \cdots + q_i(x)\hat{I}_i(x) + \cdots + q_\lambda(x)\hat{I}_\lambda(x) \quad (12)$$

in terms of so-called master integrals $\hat{I}_i(x)$ and rational functions $q_i(x) \in \mathbb{K}(x)$ where $\mathbb{K} = \mathbb{K}'(\varepsilon)$ is a rational function field over a field \mathbb{K}' containing the rational numbers as subfield; here x stands for any variable and can take over the role of x , $y := 1 - x$ or \hat{s} as introduced in (5). While the input expression may contain millions of such integrals, the output consists of much less integrals; for a concrete calculation of the form factor project one gets, e.g., $\lambda = 2506$ such master integrals. In addition, one obtains a coupled system of first-order linear differential equations in terms of the unknown master integrals. Under the assumption that the master integrals $\hat{I}_i(x)$ have a power series representation (or Laurent series representation) in x , we can apply our large moments method, that can be split into two parts:

1. Compute a large number ν of values, say $\nu = 8000$ for the quarkonic or $\nu = 20000$ for the gluonic case, of each master integral, i.e., compute $I_i(n)$ for $n = 0, \dots, \nu$ s.t.

$$\hat{I}_i(x) = \sum_{n=0}^{\nu} I_i(n)x^n + O(x^{\nu+1}). \quad (13)$$

2. Plug these expansions (13) into (12) and derive the first ν coefficients of the series expansion

$$\hat{P}(x) = \sum_{n=0}^{\nu} P(n)x^n + O(x^{\nu+1}). \quad (14)$$

Remark: Since $\hat{I}_i(x)$ and $\hat{P}(x)$ depend also on the dimensional parameter ε , also $I_i(n)$ and $P(n)$ depend on it. In a naive approach one may look for $P(n)$ and $I_i(n)$ in $\mathbb{K} = \mathbb{K}'(\varepsilon)$. However, in applications from particle physics, one seeks not only for an expansion in x but also in ε . For the three-loop case, the ε -expansion of $\hat{P}(x)$ usually starts at $1/\varepsilon^3$ and one is interested in the coefficients up to the constant term.

¹In the moment we suppress the dependence on the dimensional parameter ε .

In step 1 uncoupling algorithms implemented in the package OreSys [42] are used to obtain scalar linear differential equations in terms of the unknown master integrals \hat{I}_i of the form

$$\alpha_0(x)\hat{I}_i(x) + \alpha_1(x)D_x\hat{I}_i(x) + \cdots + \alpha_\rho(x)D_x^\rho\hat{I}_i(x) = \hat{r}(x) \quad (15)$$

where $\rho \in \mathbb{Z}_{\geq 0}$ is relatively small, say ≤ 15 , the $\alpha_i(x) \in \mathbb{K}[x]$ are polynomials in x and $\hat{r}(x)$ is given in terms of other master integrals like (12); more precisely, $\hat{r}(x)$ is again of the form (12) where one can derive the first ν coefficients of the x -expansions of the occurring master integrals either by different methods, like symbolic summation [45, 52] or by the large moments method applied recursively. Given these expansions, we compute (as in step 2 described in more details below) the x -expansion of $\hat{r}(x)$, i.e.,

$$\hat{r}(x) = \sum_{n=0}^{\nu} r(n)x^n + O(x^{\nu+1}).$$

Then by plugging in (13) into (15) and performing coefficient comparison w.r.t. x^n one obtains a linear recurrence of the form

$$\beta_0(n)I_i(n) + \beta_1(n)I_i(n+1) + \cdots + \beta_u(n)I_i(n+u) = r(n) \quad (16)$$

with polynomials $\beta_i(x) \in \mathbb{K}[x]$ and $u \in \mathbb{Z}_{\geq 0}$. Given this representation and the first u initial values (coming from other methods like symbolic summation or integration [52]), one can now compute efficiently (in linear time) the coefficients $I_i(0), \dots, I_i(\nu)$.

Remark: Note that $\mathbb{K} = \mathbb{K}'(\varepsilon)$, i.e., the coefficients β_i also depend on ε . Using the algorithms described in [20, 53] one can derive in addition the desired ε -expansions of the coefficients $I_i(n)$ up to the desired order. In [1, 16, 20–22] we address various aspects of step 1 to reduce the orders ρ and u in (15) and (16), respectively, and elaborate on various tactics to extract the ε -expansions in combination with the different uncoupling algorithms from [42].

In the following we will focus on step 2 of the method and present various improvements to gain significant speed-ups that are relevant to tackle the gluonic and quarkonic form factors. As indicated above, these improvements play also a crucial role to derive the x -expansion (and ε -expansion) of $\hat{r}(x)$ in step 1 efficiently.

Recently, we dealt in our gluonic form factor calculations with huge expressions of the form (12). For a typical example we got an expression with $\lambda = 2506$ master integrals which required about 106 GB memory in Mathematica. As it turns out, the expressions $q_i(x)$ itself are split into many smaller rational functions, say $\mu = 1213$ elements. One option would be to compute an x -expansion for each subexpression which would result in total to around a million expansions. To reduce the number of such expansions significantly, we first merged each $q_i(x)$ to one rational function by the divide-and-conquer strategy illustrated in Fig. 1. While the combining step in the inner recursions work rather efficiently using Mathematica (i.e., `MyCancel` is replaced by Mathematica's command `Cancel` or `Together`), in the outermost calls the expressions get very big. An extra complication is that the arising objects are not only rational functions in x but also depend on the dimensional parameter ε . This makes the gcd calculations extremely costly. In the final step of such a calculation

$$\begin{array}{c}
q_i(x) \\
|| \\
\frac{a_1(x)}{b_1(x)} + \cdots + \frac{a_{\lfloor \frac{\mu}{2} \rfloor}(x)}{b_{\lfloor \frac{\mu}{2} \rfloor}(x)} + \frac{a_{\lfloor \frac{\mu}{2} \rfloor + 1}(x)}{b_{\lfloor \frac{\mu}{2} \rfloor + 1}(x)} + \cdots + \frac{a_\mu(x)}{b_\mu(x)} \\
\downarrow \qquad \qquad \qquad \downarrow \\
\vdots \qquad \text{MyTogether} \qquad \qquad \vdots \qquad \text{Divide \& Conquer} \\
\downarrow \qquad \qquad \qquad \downarrow \\
\frac{A_1(x)}{B_1(x)} \qquad + \qquad \frac{A_2(x)}{B_2(x)} \\
|| \\
\frac{A_1(x)B_2(x) + A_2(x)B_1(x)}{B_1(x)B_2(x)} \\
|| \text{ MyCancel} \\
\frac{A(x)}{B(x)} \qquad \gcd(A, B) = 1
\end{array}$$

Figure 1: MyTogether

one has to deal, e.g., with the situation

$$\begin{array}{c}
A'(x, \varepsilon) = \\
\frac{\overbrace{A_1(x, \varepsilon)B_2(x) + A_2(x, \varepsilon)B_1(x, \varepsilon)}}{B_1(x, \varepsilon)B_2(x, \varepsilon)} \\
\qquad \qquad \qquad \underbrace{\hspace{10em}}_{B'(x, \varepsilon)}
\end{array}
\qquad
\begin{array}{l}
\deg_x(A') \leq 1422 \\
\deg_x(B') \leq 1405.
\end{array}$$

Then the challenge is to remove common factors in A' and B' in reasonable time. In our setting, it turns out that $B'(x, \varepsilon)$ factors sufficiently nice. E.g, we obtain factorizations of the form

$$B'(x, \varepsilon) = \text{factor}_1(x, \varepsilon)\text{factor}_2(x, \varepsilon) \dots \text{factor}_{71}(x, \varepsilon), \quad (17)$$

where the first factors are linear and the most complicated irreducible factors are of the form

$$\begin{aligned}
& 48x^4\varepsilon^6 - 9792x^3\varepsilon^6 + 19296x^2\varepsilon^6 - 19008x\varepsilon^6 + 9504\varepsilon^6 + 1006x^4\varepsilon^5 + 9264x^3\varepsilon^5 \\
& - 16896x^2\varepsilon^5 + 15264x\varepsilon^5 - 7632\varepsilon^5 - 686x^4\varepsilon^4 - 4136x^3\varepsilon^4 + 7352x^2\varepsilon^4 \\
& - 6432x\varepsilon^4 + 3216\varepsilon^4 + 95x^4\varepsilon^3 + 2416x^3\varepsilon^3 - 6664x^2\varepsilon^3 + 8496x\varepsilon^3 \\
& - 4248\varepsilon^3 - 32x^4\varepsilon^2 - 1270x^3\varepsilon^2 + 4438x^2\varepsilon^2 - 6336x\varepsilon^2 + 3168\varepsilon^2 \\
& + 25x^4\varepsilon + 330x^3\varepsilon - 1266x^2\varepsilon + 1872x\varepsilon - 936\varepsilon - 4x^4 - 32x^3 + 128x^2 - 192x + 96.
\end{aligned}$$

In order to detect candidates that may cancel, we evaluated $A'(x, \varepsilon)$, e.g., at $\varepsilon = 1234$ and use the fact that

$$\text{factor}(x, 1234) \nmid A(x, 1234) \implies \text{factor}(x, \varepsilon) \nmid A(x, \varepsilon).$$

Thus we may exclude all factors that cancel by performing much more efficient operations in $\mathbb{K}'[x]$ instead of $\mathbb{K}'[x, \varepsilon]$. For the remaining candidates we carry out polynomial division

$$A(x, \varepsilon) = q(x, \varepsilon) \text{factor}(x, \varepsilon) + r(x, \varepsilon).$$

If $r(x, \varepsilon) = 0$, we remove the factor from B and replace A by q . If $r(x, \varepsilon) \neq 0$ (which never happened), we keep the factor. Performing our divide and conquer strategy `MyTogether` with this cancellation tactic `MyCancel` we could compress the original expression $\hat{P}(x)$ from 106 GB to 5.7 GB. Note: (1) Applying the Mathematica command `Together` or `Cancel` to such a reduced expression $q_i(x)$ where the numerator and denominator are already coprime, it takes around 10 hours that Mathematica observes that nothing can be canceled. In a nutshell, utilizing the specific structure that the denominators factor nicely (see (17)) was essential to carry out the compactification in reasonable time. (2) As elaborated in [54] there are many advanced algorithms available to perform rational function arithmetic for general problems. But for our concrete situation the above approach seems to be rather optimal.

Finally, we take each simplified term in (12) and compute the expansions

$$\frac{A(x)}{B(x)} \hat{I}(x) = \sum_{n=0}^{\nu} c(n)x^n + O(x^{\nu+1}) \quad (18)$$

in parallel to get the final expansion (14). Due to step 1 we may assume that we are given the expansion (13) in x where the coefficients $I_i(n)$ are also expanded up to a certain order in ε . Thus in order to obtain the expansion in (18), we need to compute the expansion of $\frac{A(x)}{B(x)}$ in x (and also in ε). In our three-loop case the expansion of the master integrals start at order $1/\varepsilon^3$ and thus the ε -expansion of $\frac{A(x)}{B(x)}$ has to be computed only up order ε^3 to derive the desired constant term ε^0 in (18). This task can be done by simple Taylor expansion using the Mathematica command `Series`. Afterwards we carry out the expansion in x to the few coefficients of the ε -expansion. In short, it remains to compute an x -expansion of expressions of the form (18) with $A(x), B(x) \in \mathbb{K}'[x]$ being free of ε and where the first $\nu + 1$ coefficients $I_i(n) \in \mathbb{K}'$ in (13) (free of ε) are given explicitly. As it turns out, the standard command `Series` is hopeless for large values ν . As a consequence we separate the task and look first for an expansion of

$$\frac{1}{B(x)} = \sum_{n=0}^{\nu} h(n)x^n + O(x^{\nu+1}). \quad (19)$$

Here we may use the classical result (see, e.g., [55]) that the sequence $h(n)$ can be computed by a recurrence with constant coefficients that is determined by the denominator $B(x)$. However, a

typical example is

$$\begin{aligned}
 B(x) &= \overbrace{(x-2)^{26}(x-1)^{19}x^{32}(x+1)^2(2x-1)^2}^{=F_{\text{lin}}(x)} \times \\
 &\times (x^2-22x+22)^9(x^2-15x+15)^{10}(x^2-6x+6)^9(x^2-5x+5)^{10}(x^2-2x+2)^{21} \\
 &(x^2-x+1)^{19}(x^2+x-1)^{10}(x^2+2x-2)^{10}(x^2+3x-3)^{11}(x^2+4x-4)^9 \\
 &(2x^2+3x-3)^9(3x^2-14x+14)^{12}(3x^2-10x+10)(3x^2-8x+8)^{11} \\
 &(3x^2-4x+4)^8(3x^2-2x+2)^9(3x^2+x-1)^8(3x^2+2x-2)^{10}(4x^2+3x-3)^9 \\
 &(5x^2-18x+18)^{10}(5x^2-16x+16)^{11}(5x^2-2x+2)^9(5x^2+12x-12)^7 \\
 &(7x^2-6x+6)^9(11x^2+2x-2)^9(27x^2+32x-32)^9(29x^2-2x+2)^8 \\
 &(99x^2-238x+238)^{10}(x^4-8x^3+28x^2-40x+20)(x^4-6x^3+18x^2-24x+12)^8 \\
 &(x^4-4x^3+5x^2-2x+1)^9(x^4+8x^3-32x^2+48x-24)^8(3x^4+2x^3-6x^2+8x-4)^8 \\
 &(5x^4-152x^3+272x^2-240x+120)^{10}(5x^4-29x^3+27x^2+4x-2)^9 \\
 &(5x^4+16x^3-40x^2+48x-24)^{12}(5x^4+184x^3-352x^2+336x-168)^9 \\
 &(7x^4-31x^3+25x^2+12x-6)^{10}(9x^4-43x^3+37x^2+12x-6)^8 \\
 &(9x^4-11x^3+15x^2-8x+4)^8(9x^4+29x^3-15x^2-28x+14)^8 \\
 &(9x^4+80x^3-12x^2-136x+68)^9(10x^4+9x^3-103x^2+188x-94)^9 \\
 &(12x^4-85x^3+115x^2-60x+30)^{10}(13x^4-16x^3+40x^2-48x+24)^{11} \\
 &(23x^4+16x^3-40x^2+48x-24)^{11}(26x^4-83x^3-19x^2+204x-102)^{10} \\
 &\underbrace{(60x^4-79x^3-215x^2+588x-294)^8(x^6-6x^5+11x^4-8x^3-x^2+6x-2)^9}_{=F_{\text{nonlin}}(x)}
 \end{aligned}$$

with degree 1393. Consequently, the defining recurrence for $h(n)$ has also degree 1393 and one has to compute first 1393 initial values to determine the remaining values efficiently by this recurrence. To derive these values, e.g., by a Taylor expansion is by far too slow. Thus we proceed differently by applying first partial fraction decomposition. Since also here the internal Mathematica version is too slow, we derived our own version by combining well known techniques in a clever way. First, we compute polynomials $s(x)$ and $t(x)$ such that the Bezout relation

$$1 = s(x)F_{\text{nonlin}}(x) + t(x)F_{\text{lin}}(x) \quad (20)$$

holds. Here the extended Euclidean algorithm seems to be the first choice. Nevertheless, it turns out that solving the underlying linear system (related to the Sylvester matrix) is superior (by using efficient linear system solving). Using the found relation (20) and multiplying it with $\frac{1}{F_{\text{lin}}(x)F_{\text{nonlin}}(x)}$ yield the partial decomposition

$$\frac{1}{B(x)} = \frac{1}{F_{\text{lin}}(x)F_{\text{nonlin}}(x)} = \frac{s(x)}{F_{\text{lin}}(x)} + \frac{t(x)}{F_{\text{nonlin}}(x)}.$$

Next, we perform the classical method (by evaluation) to derive the partial fraction decomposition of $\frac{s(n)}{F_{\text{lin}}(n)}$. For the non-linear contribution we proceed as above by computing the corresponding Bezout relations via linear algebra. This finally leads to²

$$\frac{1}{B(x)} = \frac{a_1(x)}{(x-2)^{26}} + \frac{p_2(x)}{(x-1)^{19}} + \cdots + \frac{p_{54}(x)}{(x^6-6x^5+11x^4-8x^3-x^2+6x-2)^9}.$$

²Precisely these partial fraction decomposition techniques turned out to be useful in the summation package `Sigma` [45] for simplifying sums [46] that arise as solutions in linear difference equations [47].

In this form we finally utilize the method of recurrences with constant coefficients (with order at most 6) to calculate the series expansions of each term in parallel and finally to get the coefficients $h(n)$ in (19). To this end, we can carry out the Cauchy products to get the expansions

$$\frac{A(x)}{B(x)} \hat{I}(x) = \underbrace{\left(\sum_{n=0}^{\nu} p(n)x^n \right) \left(\sum_{n=0}^{\nu} h(n)x^n \right) \left(\sum_{n=0}^{\nu} I(n)x^n \right)}_{\text{Cauchy}} + O(x^{\nu+1}) = \sum_{n=0}^{\nu} c(n)x^n + O(x^{\nu+1}).$$

Cauchy

Also this task is challenging and further improvements can be applied as described, e.g. in [1].

4. The gluonic case

We will now discuss the progress we have made so far in the gluonic case, which requires the calculation of a much larger number of expansion coefficients at $\hat{s} = 0$. We have been able to compute enough coefficients to obtain the recursion relations associated to the following constants

$$\left\{ \zeta_2 \zeta_3, \zeta_5, \zeta_3, \ln(2)\zeta_2, \zeta_2^2, \ln^2(2)\zeta_2, \text{Li}_4\left(\frac{1}{2}\right), \ln^4(2) \right\}, \quad (21)$$

of these, only the recursions associated to $\zeta_2 \zeta_3$ and ζ_5 turned out to be solvable. In the other cases, a new feature appears compared with the quarkonic case: There are new singularities in the differential equations, which forces us to also find power-log expansions at these singularities, as well as expansions around other points. Specifically, the singularities of the differential equations associated to the constants ζ_2^2 , $\ln^2(2)\zeta_2$, $\text{Li}_4\left(\frac{1}{2}\right)$ and $\ln^4(2)$ are located at

$$\hat{s} = \{-4, -1, 3, 4\}. \quad (22)$$

In the case of the constant $\ln(2)\zeta_2$, there are also singularities at $\hat{s} = -1/2$ and $\hat{s} = 1$ in addition. We will now take the opportunity to explain in more detail the different steps of the calculation.

4.1 High energy expansion

As in the quarkonic case, we derived differential equations for the $F_{I;k,l}$'s, not only in the variable \hat{s} , but also in the variable $y = 1 - x$, from which we can then obtain the corresponding high energy expansions around $x = 0$. In some cases, the indicial equation associated to the differential equation has only integer solutions, while in other cases, also half-integer solutions are present. In the former cases, we then have power-log expansions in the standard form

$$F_{I;k,l}(x) = \sum_{i=0}^2 \sum_{j=-2}^{\infty} c_{I;k,l}(i,j) x^j \ln^i(x), \quad (23)$$

while in the latter cases, we must use \sqrt{x} as the expansion variable, so the expansions are given by

$$F_{I;k,l}(x) = \sum_{i=0}^2 \sum_{j=-2}^{\infty} c_{I;k,l}(i,j) (\sqrt{x})^j \ln^i(x). \quad (24)$$

After inserting either (23) or (24) in the differential equations in x , we obtain relations among the coefficients $c_{I;k,l}(i, j)$, leaving only a few of them undetermined. As we explained in Section 2, these undetermined coefficients can then be obtained numerically using matching conditions, see Ref. [1] for more details. However, we cannot match the expansion at $x = 0$ directly with the expansion at $\hat{s} = 0$ ($x = 1$), since, for example, in the case of the last four constants in (21), the singular points $\hat{s} = -1$ and $\hat{s} = -4$ correspond in x to $x = \frac{1}{2} \left(3 - \sqrt{5} \right) \approx 0.38$ and $x = 3 - 2\sqrt{2} \approx 0.17$, respectively. In principle, we could try to get expansions around those two points in x , but the corresponding differential equations blow up due to the square roots, and become impractical. We therefore find successive expansions at different points in \hat{s} , until we reach a value of \hat{s} large enough that we can match the corresponding expansion with the one at $x = 0$. We start by finding an expansion around the singular point $\hat{s} = -1$. In the case of the four constants in (21) under consideration, this expansion will be of the form

$$F_{I;k,l}^{(-1)}(\hat{s}) = \sum_{j=-3}^{\infty} c_{I;k,l}^{(-1)}(0, j) z^j, \quad (25)$$

where $z = \sqrt{1 + \hat{s}}$. We see that in this case we do not get logarithms and the singularity is manifested by the negative powers in z . As we will be doing for all expansions, we find a differential equation in z from the one in \hat{s} ,³ then use this differential equation to find relations among the $c_{I;k,l}^{(-1)}(i, j)$'s and the compute the coefficients left undetermined by these relations by matching with the previous expansion (in this case, the one at $\hat{s} = 0$) using a truncated version of the expansions (in this and other matchings, we used 5000 terms in the expansions, per power of the log, whenever logs were present). The radius of convergence of this expansion (25) is 1, which means we need to obtain an expansion around $\hat{s} = -2$, which will be of the form

$$F_{I;k,l}^{(-2)}(\hat{s}) = \sum_{j=0}^{\infty} c_{I;k,l}^{(-2)}(0, j) z^j, \quad (26)$$

with $z = \hat{s} + 2$. Again, the coefficients in (26) are obtained by matching the expansions at $\hat{s} = -1$ and $\hat{s} = -2$, as we did with the expansions at $\hat{s} = 0$ and $\hat{s} = -1$. Notice the absence of logs or negative powers of z in (26), which is expected since $\hat{s} = -2$ is not a singular point. Since the closest singularity is at $\hat{s} = -1$ the radius of convergence of the expansion at $\hat{s} = -2$ is again 1. We then repeat the process at $\hat{s} = -3$, where, again, the radius of convergence is 1 and the expansion is similar to (26). We then match the expansion at $\hat{s} = -3$ with an expansion at $\hat{s} = -4$, which is of the form

$$F_{I;k,l}^{(-4)}(\hat{s}) = \sum_{i=0}^1 \sum_{j=-2}^{\infty} c_{I;k,l}^{(-4)}(i, j) z^j \ln^i(z), \quad (27)$$

where $z = s + 4$. After the point $\hat{s} = -4$, all expansions will be regular and similar to (26). Since the nearest singularity is $\hat{s} = -1$, the radius of convergence of the expansion (27) is 3, so we can now match with an expansion at $\hat{s} = -7$. The radius of convergence of this expansion will also be 3, and we can then match it with an expansion at $\hat{s} = -10$, which radius of convergence will be

³One of the solutions to the indicial equation of this differential equation is $1/2$. Hence the square root in the change of variables from z to \hat{s} . This usually happens (although not always) around singular points.

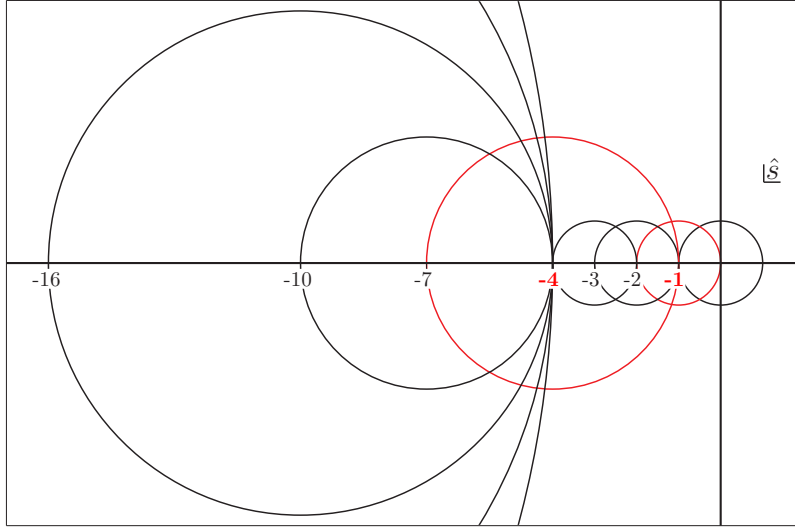


Figure 2: Radii of convergence of the successive points around which expansions are matched in the case of the last four constants in (21). The points $\hat{s} = -1$ and $\hat{s} = -4$, shown in red, are singularities of the differential equations.

6, and can therefore be matched with an expansion at $\hat{s} = -16$, and so on. We see that since the closest singularity for $\hat{s} < -4$ is the singularity at $\hat{s} = -4$, we can match expansions with radii of convergence that increase geometrically by a factor of two. We therefore do consecutive matchings of expansions at the following points:

$$\hat{s} = \{-196, -100, -52, -28, -16, -10, -7, -4, -3, -2, -1\}, \quad (28)$$

in reverse order. A few of these points, together with their corresponding convergence disks are shown in Fig. 2. In the case of the constant $\ln(2)\zeta_2$, we also need to match at $\hat{s} = -1/2$ and $\hat{s} = -3/2$.

The point $s = -196$ corresponds in x to $x = (99 + 70\sqrt{2})^{-1} \approx 0.005$, which is small enough to allow us to match this expansion with the expansion at $x = 0$. We did this for all cases available and found all coefficients left unconstrained by the differential equations with a precision of around a thousand digits. This was enough to obtain all coefficients using the PSLQ algorithm in terms of known constants, which in the cases under consideration were limited to the set $\{1, \pi, \zeta_2\}$, as expected since we are dealing at the moment with expansions associated to the weight 4 constants in (21), and therefore, the weight of any new constants on top of them cannot be higher than 2. Odd powers of π only appear in the imaginary parts of the coefficients, which should cancel in the final result for the form factors.

4.2 The $\hat{s} = 4$ threshold

Let us focus again on the last four constants in (21). Since the differential equations have singularities at $\hat{s} = 3$ and $\hat{s} = 4$, we again need to match consecutive expansions until we reach the $\hat{s} = 4$ threshold. As we already mentioned, the radius of convergence of the expansion around $\hat{s} = 0$

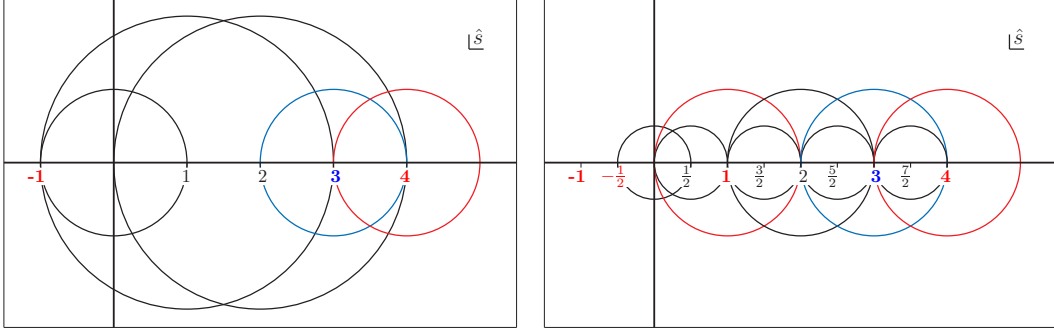


Figure 3: Radii of convergence of the successive points around which expansions are matched in order to obtain the threshold expansion at $\hat{s} = 4$ (shown in red). The point $\hat{s} = 3$ (shown in blue) is singular according to the differential equations, but the singularity disappears due to initial conditions. Left panel: radii of convergence of the points used in the case of the last four constants in (21). Right panel: radii of convergence of the points used in the case of $\ln(2)\zeta_2$.

is 1, so we need to match with an expansion around $\hat{s} = 1$. The radius of convergence of the latter expansion is now 2, since the closer singularity is at $\hat{s} = -1$, so in principle we could match this expansion directly with the one at $\hat{s} = 3$. However, we found better precision by matching first with an expansion around $\hat{s} = 2$ before reaching $\hat{s} = 3$. The expansions around $\hat{s} = 3$ are of the form

$$F_{I;k,l}^{(3)}(\hat{s}) = \sum_{j=-3}^{\infty} c_{I;k,l}^{(3)}(0, j) z^j, \quad (29)$$

where $z = \sqrt{3 - \hat{s}}$. We see that the singularity is manifested in terms of negative powers of z . However, it is interesting to notice that although the differential equations allow us to search for such type of solutions, once we match with the expansion at $\hat{s} = 2$, all coefficients of the negative powers in (29) turn out to be effectively zero, and therefore the singularity disappears.

Finally, we match the expansions at $\hat{s} = 3$ and $\hat{s} = 4$. The latter has the following form

$$F_{I;k,l}^{(4)}(\hat{s}) = \sum_{i=0}^1 \sum_{j=-3}^{\infty} c_{I;k,l}^{(4)}(i, j) z^j \ln^i(z), \quad (30)$$

where $z = \sqrt{4 - \hat{s}}$. All of the expansions from $\hat{s} = 1$ to $\hat{s} = 4$ were matched using 10000 expansion terms (per power of the log). These expansion points and their corresponding convergence disks are depicted in left panel of Fig. 3. On the right panel, we show the corresponding convergence disks in the case of the constant $\ln(2)\zeta_2$, where we also have a singularity at $\hat{s} = 1$. In this case, in order to improve the precision of the expansion coefficients at $\hat{s} = 4$, we matched successive expansions at every integer and half-integer value of \hat{s} . The corresponding radii of convergence are shown on the right panel of Fig. 3.

We were able to determine the coefficients of the expansions at $\hat{s} = 4$ with a precision of around 1400 decimal places. Again, we tried to use the PSLQ algorithm to try to express the coefficients in terms of known constants. Much like in the quarkonic case, not all coefficients could be determined this way, but we could reduce the number of undetermined coefficients to just 5 in total in the case of the last four constants in (21).

5. Conclusions

We have made substantial progress applying the large moments method to the calculation of massive three-loop non-singlet QCD form factors. This method allows us to find recursions and differential equations associated to the constants appearing in the corresponding low energy expansions. The recursion relations can then be used to obtain an even larger number of coefficients than the ones used to derive them, which allows us to get very precise numerical evaluations within the radius of convergence of the expansions. The differential equations can then be used to obtain expansions at other points, which can be expressed in terms of just few of the coefficients. These coefficients can then be computed numerical by matching the expansions at intermediate points. In this way, we were able to find expansions around all kinematic points of interest. In the gluonic case, this procedure turns out to be somewhat more complicated than in the quarkonic case, since the differential equations have additional spurious singularities (they must cancel in the final physical result), which forces us to obtain successive expansions at many intermediate points. So far we have been able to apply this method to five of the constants appearing the expansion at $\hat{s} = 0$ ($x = 1$), which required only 1000 coefficients in the case of the last four constants in (21), around 6000 coefficients in the case of $\ln(2)\zeta_2$, and no more than 14500 coefficients in the case of ζ_3 . The remaining constants require a much larger number of coefficients, which are being computed at the moment. Once we obtain them, we will be able to proceed as in the other cases.

Acknowledgment. This work has been supported in part by the Austrian Science Fund (FWF) 10.55776/P33530.

References

- [1] J. Blümlein, A. De Freitas, P. Marquard, N. Rana and C. Schneider, *Phys. Rev. D* **108** (2023) no.9, 094003 [arXiv:2307.02983 [hep-ph]].
- [2] W. Bernreuther, R. Bonciani, T. Gehrmann, R. Heinesch, T. Leineweber, P. Mastrolia, and E. Remiddi, *Two-loop QCD corrections to the heavy quark form-factors: The Vector contributions*, *Nucl. Phys. B* **706** (2005) 245–324 [hep-ph/0406046].
- [3] W. Bernreuther, R. Bonciani, T. Gehrmann, R. Heinesch, T. Leineweber, P. Mastrolia, and E. Remiddi, *Two-loop QCD corrections to the heavy quark form-factors: Axial vector contributions*, *Nucl. Phys. B* **712** (2005) 229–286 [hep-ph/0412259].
- [4] W. Bernreuther, R. Bonciani, T. Gehrmann, R. Heinesch, T. Leineweber, and E. Remiddi, *Two-loop QCD corrections to the heavy quark form-factors: Anomaly contributions*, *Nucl. Phys. B* **723** (2005) 91–116 [hep-ph/0504190].
- [5] W. Bernreuther, R. Bonciani, T. Gehrmann, R. Heinesch, P. Mastrolia, and E. Remiddi, *Decays of scalar and pseudoscalar Higgs bosons into fermions: Two-loop QCD corrections to the Higgs-quark-antiquark amplitude*, *Phys. Rev. D* **72** (2005) 096002 [hep-ph/0508254].
- [6] J. Gluza, A. Mitov, S. Moch, and T. Riemann, *The QCD form factor of heavy quarks at NNLO*, *JHEP* **07** (2009) 001 [arXiv:0905.1137 [hep-ph]].

- [7] J. Ablinger, A. Behring, J. Blümlein, G. Falcioni, A. De Freitas, P. Marquard, N. Rana and C. Schneider, *Heavy quark form factors at two loops*, Phys. Rev. D **97** (2018) no.9, 094022 [arXiv:1712.09889 [hep-ph]].
- [8] J.M. Henn, A.V. Smirnov, and V.A. Smirnov, *Analytic results for planar three-loop integrals for massive form factors*, JHEP **12** (2016) 144 [arXiv:1611.06523 [hep-ph]].
- [9] J. Henn, A.V. Smirnov, V.A. Smirnov and M. Steinhauser, *Massive three-loop form factor in the planar limit*, JHEP **1701** (2017) 074 [arXiv:1611.07535 [hep-ph]].
- [10] J. Ablinger, J. Blümlein, P. Marquard, N. Rana and C. Schneider, *Heavy Quark Form Factors at Three Loops in the Planar Limit* Phys. Lett. B **782** (2018) 528–532 [arXiv:1804.07313 [hep-ph]].
- [11] T. Ahmed, J.M. Henn and M. Steinhauser, *High energy behaviour of form factors*, JHEP **06** (2017) 125 [arXiv:1704.07846 [hep-ph]].
- [12] J. Blümlein, P. Marquard and N. Rana, *Asymptotic behavior of the heavy quark form factors at higher order* Phys. Rev. D **99** (2019) no.1, 016013 [arXiv:1810.08943 [hep-ph]].
- [13] R.N. Lee, A.V. Smirnov, V.A. Smirnov and M. Steinhauser, *Three-loop massive form factors: complete light-fermion corrections for the vector current*, JHEP **03** (2018) 136 [arXiv:1801.08151 [hep-ph]].
- [14] R.N. Lee, A.V. Smirnov, V.A. Smirnov and M. Steinhauser, *Three-loop massive form factors: complete light-fermion and large- N_c corrections for vector, axial-vector, scalar and pseudo-scalar currents*, JHEP **05** (2018) 187 [arXiv:1804.07310 [hep-ph]].
- [15] E. Remiddi and J. A. M. Vermaseren, *Harmonic polylogarithms*, Int. J. Mod. Phys. A **15** (2000) 725-754 [arXiv:hep-ph/9905237 [hep-ph]].
- [16] J. Blümlein, P. Marquard, N. Rana and C. Schneider, *The Heavy Fermion Contributions to the Massive Three Loop Form Factors*, Nucl. Phys. B **949** (2019) 114751 [arXiv:1908.00357 [hep-ph]].
- [17] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *Singlet and nonsinglet three-loop massive form factors*, Phys. Rev. D **106** (2022) 034029 [arXiv:2207.00027 [hep-ph]].
- [18] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *Massive Vector Form Factors to Three Loops*, Phys. Rev. Lett. **128** (2022) 172003 [arXiv:2202.05276 [hep-ph]];
K. Schönwald, *Massive form factors at $O(\alpha_s^3)$* , PoS (LL2022) 006 [arXiv:2207.06705 [hep-ph]].
- [19] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *Massive three-loop form factors: anomaly contribution*, Phys. Rev. D **107** (2023) no.9, 094017 [arXiv:2302.00693 [hep-ph]].
- [20] J. Blümlein and C. Schneider, *The Method of Arbitrarily Large Moments to Calculate Single Scale Processes in Quantum Field Theory*, Phys. Lett. B **771** (2017) 31–36, [arXiv:1701.04614 [hep-ph]].

- [21] J. Ablinger, J. Blümlein, P. Marquard, N. Rana and C. Schneider, *Automated Solution of First Order Factorizable Systems of Differential Equations in One Variable*, Nucl. Phys. B **939** (2019) 253–291. [arXiv:1810.12261 [hep-ph]].
- [22] J. Ablinger, C. Schneider, A. Behring, J. Blümlein, A. De Freitas, *Algorithms to solve coupled systems of differential equations in terms of power series*, PoS (LL2016) 005, [arXiv:1608.05376 [cs.SC]].
- [23] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. Goedicke, A. von Manteuffel, C. Schneider, and K. Schönwald. *The unpolarized and polarized single-mass three-loop heavy flavor operator matrix elements $A_{gg,Q}$ and $\Delta A_{gg,Q}$* , JHEP **12** (2022) 134 [arXiv:2211.05462 [hep-ph]].
- [24] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, C. Schneider and K. Schönwald, *The first-order factorizable contributions to the three-loop massive operator matrix elements $A_{Qg(3)}$ and $\Delta A_{Qg(3)}$* , Nucl. Phys. B **999** (2024) 116427 [arXiv:2311.00644 [hep-ph]].
- [25] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, C. Schneider and K. Schönwald, *The non-first-order-factorizable contributions to the three-loop single-mass operator matrix elements $A_{Qg}^{(3)}$ and $\Delta A_{Qg}^{(3)}$* , Phys. Lett. B **854** (2024) 138713 [arXiv:2403.00513 [hep-ph]].
- [26] J. Blümlein, P. Marquard, C. Schneider, and K. Schönwald. *The massless three-loop Wilson coefficients for the deep-inelastic structure functions F_2 , F_L , xF_3 and g_1* , JHEP **11** (2022) 156 [arXiv:2208.14325 [hep-ph]].
- [27] J. Blümlein, P. Marquard, C. Schneider and K. Schönwald, *The three-loop unpolarized and polarized non-singlet anomalous dimensions from off shell operator matrix elements*, Nucl. Phys. B **971** (2021) 115542 [arXiv:2107.06267 [hep-ph]].
- [28] 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*, Comput. Phys. Commun. **180** (2009) 2143–2165 [arXiv:0902.4091 [hep-ph]].
- [29] P. Nogueira, *Automatic Feynman graph generation*, J. Comput. Phys. **105** (1993) 279–289.
- [30] J.A.M. Vermaseren, *New features of FORM*, math-ph/0010025.
- [31] M. Tentyukov and J.A.M. Vermaseren, *The Multithreaded version of FORM*, Comput. Phys. Commun. **181** (2010) 1419–1427 [hep-ph/0702279].
- [32] T. van Ritbergen, A. N. Schellekens and J. A. M. Vermaseren, *Group theory factors for Feynman diagrams*, Int. J. Mod. Phys. A **14** (1999) 41–96 [hep-ph/9802376].
- [33] P. Marquard and D. Seidel, *The Crusher algorithm*, unpublished.
- [34] J. Ablinger, J. Blümlein and C. Schneider, *Harmonic Sums and Polylogarithms Generated by Cyclotomic Polynomials*, J. Math. Phys. **52** (2011) 102301 [arXiv:1105.6063 [math-ph]].

- [35] J.A.M. Vermaseren, *Harmonic sums, Mellin transforms and integrals*, Int. J. Mod. Phys. A **14** (1999) 2037–2076 [hep-ph/9806280].
- [36] J. Blümlein and S. Kurth, *Harmonic sums and Mellin transforms up to two loop order*, Phys. Rev. D **60** (1999) 014018 [hep-ph/9810241].
- [37] J. Ablinger, J. Blümlein and C. Schneider, *Analytic and Algorithmic Aspects of Generalized Harmonic Sums and Polylogarithms*, J. Math. Phys. **54** (2013) 082301 [arXiv:1302.0378 [math-ph]].
- [38] J. Ablinger, J. Blümlein, C.G. Raab and C. Schneider, *Iterated Binomial Sums and their Associated Iterated Integrals*, J. Math. Phys. **55** (2014) 112301 [arXiv:1407.1822 [hep-th]].
- [39] J. Blümlein, *Algebraic relations between harmonic sums and associated quantities*, Comput. Phys. Commun. **159** (2004) 19–54 [hep-ph/0311046].
- [40] J. Ablinger, *Computer Algebra Algorithms for Special Functions in Particle Physics*, Ph.D. Thesis, Linz U. (2012) [arXiv:1305.0687[math-ph]].
- [41] J. Ablinger, *The package HarmonicSums: Computer Algebra and Analytic aspects of Nested Sums*, PoS (LL2014) 019 [arXiv:1407.6180 [cs.SC]].
- [42] S. Gerhold. *Uncoupling systems of linear Ore operator equations*, Master’s thesis, RISC, J. Kepler University Linz (2002).
- [43] J. Ablinger, J. Blümlein, A. De Freitas, M. van Hoeij, E. Imamoglu, C. G. Raab, C. S. Radu and C. Schneider, *Iterated Elliptic and Hypergeometric Integrals for Feynman Diagrams*, J. Math. Phys. **59** (2018) no.6, 062305 doi:10.1063/1.4986417 [arXiv:1706.01299 [hep-th]].
- [44] M. Kauers, M. Jaroschek, and F. Johansson, *Ore Polynomials in Sage*, in: Computer Algebra and Polynomials, Editors: J. Gutierrez, J. Schicho, Josef, M. Weimann, Eds.. Lecture Notes in Computer Science **8942** (Springer, Berlin, 2015) 105–125 [arXiv:1306.4263 [cs.SC]].
- [45] C. Schneider. *Symbolic Summation Assists Combinatorics*. Sem. Lothar. Combin. **56** (2007) 1–36. Article B56b.
- [46] C. Schneider. *Refined telescoping algorithms in $R\Pi\Sigma$ -extensions to reduce the degrees of the denominators*. In G. Jeronimo, editor, *ISSAC ’23: Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, pages 498–507. ACM, 2023. [arXiv:2302.03563 [cs.SC]].
- [47] S. A. Abramov, M. Bronstein, M. Petkovšek, and C. Schneider. *On Rational and Hypergeometric Solutions of Linear Ordinary Difference Equations in $\Pi\Sigma^*$ -field extensions*, J. Symb. Comput. **107** (2021) 23–66 [arXiv:2005.04944 [cs.SC]].
- [48] H. Ferguson and R. Forcade, *Generalization of the Euclidean algorithm for real numbers to all dimensions higher than two*, Bull. Am. Math. Soc. **1** (1979) 912–914;
H. Ferguson and D. Bailey, *A Polynomial Time, Numerically Stable Integer Relation Algorithm*, Technical Report RNR-91-032 (1991).

- [49] D.H. Bailey and D.J. Broadhurst, *Parallel integer relation detection: Techniques and applications*, Math. Comput. **70** (2001) 1719–1736 [arXiv:math/9905048 [math.NA]].
- [50] K. Chetyrkin and F. Tkachov. *Integration by parts: The algorithm to calculate beta functions in 4 loops*, Nucl. Phys. B **192** (1981) 159–204.
- [51] S. Laporta. *High precision calculation of multiloop Feynman integrals by difference equations*, Int. J. Mod. Phys. A **15** (2000) 5087–5159 [arXiv:hep-ph/0102033 [hep-ph]].
- [52] J. Blümlein, C. Schneider. *The SAGEX Review on Scattering Amplitudes, Chapter 4: Multi-loop Feynman Integrals*, J. Phys. A: Math. Theor. **55**(44) (2022), 1-37 [arXiv:2203.13015 [hep-th]].
- [53] J. Blümlein, S. Klein, C. Schneider, F. Stan. *A Symbolic Summation Approach to Feynman Integral Calculus*, J. Symbolic Comput. **47** (2012) 1267-1289 [arXiv:1011.2656 [cs.SC]].
- [54] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, third edition, 2013.
- [55] M. Kauers and P. Paule. *The concrete tetrahedron*. Texts and Monographs in Symbolic Computation. (Springer, Wien, 2011).