

The Method of Arbitrarily Large Moments to Calculate Single Scale Processes in Quantum Field Theory

Johannes Blümlein¹ and Carsten Schneider²

¹*Deutsches Elektronen-Synchrotron, DESY, Platanenallee 6, D-15738 Zeuthen, Germany*

²*Research Institute for Symbolic Computation (RISC),*

Johannes Kepler University Linz, Altenbergerstraße 69, A-4040 Linz, Austria

We devise a new method to calculate a large number of Mellin moments of single scale quantities using the systems of differential and/or difference equations obtained by integration-by-parts identities between the corresponding Feynman integrals of loop corrections to physical quantities. These scalar quantities have a much simpler mathematical structure than the complete quantity. A sufficiently large set of moments may even allow the analytic reconstruction of the whole quantity considered, holding in case of first order factorizing systems. In any case, one may derive highly precise numerical representations in general using this method, which is otherwise completely analytic.

PACS numbers: 12.20.-m, 12.38.-t, 03.70.+k, 13.85.-t, 25.30.-c

INTRODUCTION

Single scale higher order QED and QCD calculations in the massless [1, 2] and massive [3] cases at fixed Mellin moment n are given by polynomials of rational numbers and a series of a few special constants, the multiple ζ -values, and possible generalizations thereof [4]. This is irrespectively the case, whether or not the functional representation for general values of n obeys an equation factorizing in first order, [1], or not [5]. If one has access [6] to an algorithm, through which a large number of moments, e.g. $N = 2000$ or larger, can be calculated, the method of guessing, cf. [8], for holonomic problems, which often appear in physics applications, allows one to gain one large difference equation describing the corresponding problem [9]. If this equation is solvable in difference ring theory [10] one finds the solution for general values of n for this quantity without any further assumptions, e.g. made in [2, 11]. In any case, significantly more moments will allow one to constrain the considered quantity much better numerically using approximation methods, e.g. through Chebyshev-polynomials or other interpolation methods.

In the following we describe an algorithm through which the system of differential equations, or associated to it, that of difference equations, available by the integration-by-parts relations [12], can be used to compute a large number of Mellin moments for the master integrals, and through them for the whole problem. The solution of the associated difference equations will need a relatively low number of initial values which have to be provided. The corresponding sequences of rational numbers mentioned above actually form the problematic part in gaining the general n result from the moments, since very involved, and in some cases yet unknown, functions span the corresponding sequences. In any case, the

method allows either to find the one-dimensional distribution from a large but finite amount of moments, or at least to constrain it numerically at high accuracy.

We will give a brief illustration of the algorithm in case of a system of massive 3-loop master integrals, for which first a larger number of moments is generated, corresponding difference equations are found and solved for general values of the Mellin variable n .

THE ALGORITHM

Single scale master integrals can be represented as analytic functions $\hat{I}_i(x) = \sum_{n=0}^{\infty} I_i(n)x^n$ with $1 \leq i \leq m$. We aim at computing a large number of coefficients $I_i(0), I_i(1), \dots, I_i(s)$. Usually the coefficients $I_i(n)$ depend on the dimensional parameter ε which itself can be expanded in a Laurent series in ε of a certain order $o \in \mathbb{Z}$. We are interested in calculating the coefficients $I_i^{(k)}(n) \in \mathbb{R}$ of the expansions

$$\hat{I}_i(x) = \sum_{n=0}^{\infty} I_i(n)x^n = \sum_{n=0}^{\infty} \left(\sum_{k=o}^{\infty} I_i^{(k)}(n)\varepsilon^k \right) x^n \quad (1)$$

up to a certain degree in ε , $t_i \in \mathbb{Z}$. More precisely, we want to compute for $1 \leq i \leq m$ the initial values

$$I_i^{(k)}(0), I_i^{(k)}(1), I_i^{(k)}(2), \dots, I_i^{(k)}(s) \in \mathbb{R} \quad (2)$$

for the ε -orders $o \leq k \leq t_i$.

In our approach we rely on the property that these unknown functions $\hat{I}_i(x)$ are usually described by a coupled system of first order linear differential equations

$$D_x \begin{pmatrix} \hat{I}_1(x) \\ \hat{I}_2(x) \\ \dots \\ \hat{I}_m(x) \end{pmatrix} = A \begin{pmatrix} \hat{I}_1(x) \\ \hat{I}_2(x) \\ \dots \\ \hat{I}_m(x) \end{pmatrix} + \begin{pmatrix} \hat{r}_1(x) \\ \hat{r}_2(x) \\ \dots \\ \hat{r}_m(x) \end{pmatrix} \quad (3)$$

with $D_x = d/dx$ and where A is an $m \times m$ matrix with entries consisting of polynomials in ε and x and the entries \hat{r}_i can be given in form of the expansions

$$\hat{r}_i(x) = \sum_{n=0}^{\infty} r_i(n)x^n = \sum_{n=0}^{\infty} \left(\sum_{k=0}^{\infty} r_i^{(k)}(n)\varepsilon^k \right) x^n \quad (4)$$

with $r_i^{(k)}(n) \in \mathbb{R}$ for $n \in \mathbb{N}$. Another important assumption is that the coefficients $r_i^{(k)}(n)$ can be determined efficiently by using either the method under consideration in a recursive fashion or by using, e.g., symbolic summation and integration techniques [10, 13]. Further, we assume that for reasonable small numbers s'_i the coefficients $I_i^{(k)}(n)$ with $0 \leq n \leq s'_i$ can be computed up to certain ε -degrees t'_i , i.e., $0 \leq k \leq t'_i$ as a preprocessing step. Given this input we introduce the following efficient algorithm that computes the coefficients (2) up to large values of n .

- (1) Using decoupling algorithms [14, 15] transform the system (3) symbolically to one scalar linear differential equation

$$\sum_{k=0}^m a_k(x, \varepsilon) D_x^{(k)} \hat{I}_1(x) = \sum_{i=1}^m d_i(x, \varepsilon) \hat{r}_i(x), \quad (5)$$

where the $a_i(x, \varepsilon)$ and $r_i(x)$ are polynomials in x and ε ; in addition, one obtains identities of the form

$$\begin{aligned} \hat{I}_i(x) &= \sum_{j=2}^m \sum_k e_{i,j,k}(x, \varepsilon) D_x^k \hat{I}_j(x) \\ &+ \sum_{j=1}^m \sum_k f_{i,j,k}(x, \varepsilon) D_x^k \hat{r}_j(x) \end{aligned} \quad (6)$$

for $2 \leq i \leq n$ where the $e_{i,j,k}(x, \varepsilon)$ and $f_{i,j,k}(x, \varepsilon)$ are rational functions in x and ε .

The algorithm proceeds now as follows: Compute in steps (2)-(4) the initial values for $\hat{I}_1(x)$ by using the scalar differential equation (5), and compute afterwards in step (5) the initial values for the remaining integrals with the given formulas (6).

- (2) Plug $\hat{I}_1(x) = \sum_{n=0}^{\infty} I_1(n)x^n$ and $\hat{r}_i(x) = \sum_{n=0}^{\infty} r_i(n)x^n$ into (5) and eliminate D_x by using the property $D_x \sum_{n=0}^{\infty} h(n)x^n = \sum_{n=1}^{\infty} n h(n)x^{n-1}$ for a power series $\sum_{n=0}^{\infty} h(n)x^n$. Then by coefficient comparison w.r.t. x^n and using appropriate shifts one gets a linear recurrence of the form

$$\sum_{k=0}^d b_k(n, \varepsilon) I_1(n+k) = \rho(n, \varepsilon), \quad (7)$$

with $\rho(n, \varepsilon) = \sum_{j=1}^m \sum_{k=0}^l g_{j,k}(n, \varepsilon) r_j(n+k)$ for some $l \in \mathbb{N}$ where the $b_k(n, \varepsilon)$ and $g_{j,k}(n, \varepsilon)$ are

polynomials in n and ε . Finally, divide the equation by a factor ε^u for some $u \geq 0$ in order to obtain updated polynomials $b_k(n, \varepsilon)$ in n, ε where not all $b_k(n, 0)$ with $0 \leq k \leq d$ are zero; the $g_{j,k}(n, \varepsilon)$ are now polynomials in n and Laurent polynomials in ε .

- (3) We write the right hand side of (7) in the expanded representation

$$\rho(n, \varepsilon) = \sum_{k=0}^{\infty} \rho^{(k)}(n) \varepsilon^k, \quad (8)$$

with $\rho^{(k)}(n) \in \mathbb{R}$. Since the $r_i^{(k)}(n) \in \mathbb{R}$ in (4) can be computed efficiently by assumption, the coefficients $\rho^{(k)}(n)$ for sufficiently large $k \in \mathbb{Z}$ and $n \in \{0, 1, \dots, s\}$ can be obtained explicitly without any cost.

- (4) We proceed as follows; compare [16], Lemma 1. Plugging

$$I_1(n) = \sum_{k=0}^{\infty} I_1^{(k)}(n) \varepsilon^k \quad (9)$$

into (7) and doing coefficient comparison w.r.t. ε^o yield the constraint

$$\sum_{k=0}^{d'} b_k(n, 0) I_1^{(o)}(n+k) = \rho^{(o)}(n)$$

for some $d' \leq d$ with $b_{d'}(n) \neq 0$. Choose $\delta \in \mathbb{N}$ such that $b_{d'}(n) \neq 0$ for all $n \in \mathbb{N}$ with $n \geq \delta$. Then we can compute with the first values $I_1^{(o)}(0), \dots, I_1^{(o)}(n+d'+\delta-1)$ and the rule

$$I_1^{(o)}(n) \leftarrow \frac{\rho^{(o)}(n') - \sum_{k=0}^{d'-1} b_k(n', 0) I_1^{(o)}(n'+k)}{b_{d'}(n', 0)} \quad (10)$$

for $n' = n - d', n \geq d' + \delta$ all the other values in linear time. Now insert (9) with the explicitly computed values $I_1^{(o)}(n)$ with $0 \leq n \leq s$ into (7) and move these given values to the right hand side. This yields

$$\sum_{k=0}^{d'} b_k(n, \varepsilon) I_1'(n+k) = \sum_{k=o+1}^{\infty} \rho'^{(k)}(n) \varepsilon^k$$

for $I_1'(n) = \sum_{k=o+1}^{\infty} I_1^{(k)}(n)$ where the $\rho'^{(k)}(n)$ for $0 \leq n \leq s$ and sufficiently large k are given explicitly. Now we are in the position to repeat this tactic iteratively to compute the remaining coefficients: next $I_1^{o+1}(n)$ with $0 \leq n \leq s$, afterwards $I_1^{o+2}(n)$ with $0 \leq n \leq s$, and eventually $I_1^{t_i}(n)$ with $0 \leq n \leq s$.

- (5) Now expand the rational functions $f_{i,j,k}(x, \varepsilon)$ and $g_{i,j,k}(x, \varepsilon)$ in (6) in a Laurent-series expansion w.r.t. ε and compute their coefficients as power series in x up to the necessary orders. Finally, combine all the explicitly given (finite) expansions in (6) using component-wise addition and Cauchy-products which yield the coefficients $I_i^{(k)}(n)$ for $2 \leq i \leq m$, $0 \leq k \leq t_i$ and $0 \leq n \leq s$.

The following remarks are in order; related considerations have been applied also to our algorithms to solve coupled systems in terms of nested sums over hypergeometric products [13, 17–19].

- (i) Our algorithm requires that sufficiently many initial values/coefficients of $\hat{r}_i(x)$ and $\hat{I}_1(x)$ up to the right ε -order are computed as a preprocessing step. The necessary bounds for these numbers can be determined by analyzing the formulas (7) and (6) accordingly.
- (ii) For simplicity, we assumed that only one scalar differential equation arises, as it happens in most examples. In general, several scalar differential equations might arise, i.e., steps (2)–(4) have to be executed several times.
- (iii) One can choose any λ with $1 \leq \lambda \leq m$ to determine a scalar differential equation in $\hat{I}_\lambda(x)$. Different choices of λ might lead to different recurrences (7) with different orders d and formulae (6) of different size. In our calculations we analyze for each λ ($1 \leq \lambda \leq m$) the obtained symbolic formulae and choose that λ for the calculation steps (3)–(5) that serves us best, e.g., to minimize the required ε -orders for the $\hat{r}_i(x)$ or minimize the the recurrence order d .
- (iv) Often it is a challenge to determine the first initial values of $I_i^k(n)$ to activate the recurrence formula (10). Thus it is highly desirable to keep the order d in (7) as small as possible. Since d gets smaller if the degrees of the $a_i(x, \varepsilon)$ w.r.t. x in (5) can be made smaller, the following refinement can be applied. Divide (5) by the polynomial $h = \gcd_x(a_0, \dots, a_n)$ in x . In most applications this reduces the degrees of the polynomials $a_i(x, \varepsilon)$ w.r.t. x heavily and thus produces recurrences with much lower order, e.g., $d = 4$ instead of $d = 20$ in (7). The price to be paid is to determine the coefficients $\rho^{(k)}(n)$ in (8) by extracting the n th coefficients from the right hand side of (5). Here the more involved calculation steps similarly as given in step (5) must be carried out.

All these aspects have been implemented within our package `SolveCoupledSystem` [13, 18, 19] which uses subroutines of the summation package `Sigma` [10] and the uncoupling package `OreSys` [15].

	ε^{-3}	ε^{-2}	ε^{-1}	ζ_2	ε^0	ζ_2	ζ_3
J_1	8	24	48	8	89	24	8
J_2	8	23	42	8	81	21	8
J_3	-	8	19	-	63	-	5

TABLE I. The number of moments needed to find the associated difference equations.

	ε^{-3}		ε^{-2}		ε^{-1}		ζ_2		ε^0		ζ_2		ζ_3	
	O	R	O	R	O	R	O	R	O	R	O	R	O	R
J_1	1	2	2	5	3	11	1	2	4	20	2	5	1	2
J_2	1	2	2	5	3	10	1	2	4	18	2	5	1	2
J_3	-	-	1	2	2	4	-	-	4	12	-	-	1	1

TABLE II. The order and degree of the associated difference equations.

AN EXAMPLE

Let us consider a set of three 3-loop master integrals $\{\hat{J}_1(x, \varepsilon), \hat{J}_2(x, \varepsilon), \hat{J}_3(x, \varepsilon)\}$ which obey a 3×3 inhomogeneous linear coupled system, cf. (3), as an illustrative example. They contribute to the massive operator matrix element $A_{gg}^{(3)}$ [20], and are given in Eqs. (3.62–3.64). In a first step, we will use our above method to compute a large series of fixed moments for $J_{1,2,3}(n, \varepsilon)$, and in a second step we will use this information to guess recurrences for these functions. In our example the integrals J_i have poles up to ε^{-3} and we would like to compute the moments up to the constant term $O(\varepsilon^0)$. Since the number of moments n_0 needed to guess a recurrence is not known a priori, the moments are computed up to a reasonable number. If this is not sufficient, further moments can be computed reusing the previous calculations. E.g., using our algorithm the time to generate $n_0 = 500$ moments amounts to 48 sec and $n_0 = 2000$ moments require 569 sec in the present example. In a nutshell, higher moments can therefore easily be produced.

The guessing method to find the minimal difference equation requires the numbers n_0 , depending on ε and whether the respective set corresponds to the purely rational term or a respective contribution $\propto \zeta_i$. The values are summarized in Table I. Here the highest number turns out to be $n_0 = 89$. The guessing method [8] yields a difference equation

$$\sum_{k=0}^O a_k(n)F(n+k) = 0 \quad (11)$$

for each power in ε . Here O and R denote the order and degree of the difference equation, i.e. the number of shift-operators and the maximal degree of the polynomials $a_k(n)$. The orders and degrees of the different difference equations are listed in Table II. The number of moments always includes a sufficiently large safety mar-

gin to validate the corresponding difference equations, which may even be enlarged. We generate additional moments according to their order to provide the needed initial values to solve the difference equations.

In the present case, using the algorithm of Ref. [13], all difference equations can be solved completely using difference ring theory [10]. Besides rational terms in n the result for $J_1(n)$ and $J_2(n)$ is spanned by the harmonic sums [21] up to $S_{1,2}(n)$. $J_3(n)$, furthermore, contains the finite binomial sum

$$\frac{1}{4^n} \binom{2n}{n} \left\{ \sum_{k=1}^n \frac{4^k S_1(k-1)}{\binom{2k}{k} k^2} - 7\zeta_3 \right\}, \quad (12)$$

multiplied by a rational term in n , see also [22]. All appearing letters building the nested sums forming the master integrals are found automatically. The above example is a simpler one. Let us mention that, for comparison, the computational time for the complete pole terms of more involved massive 3-loop examples requires the knowledge of about ~ 1000 even moments [9] and is estimated to amount to several weeks [23].

In case the difference equations would turn out not to be first order factorizable using the algorithm of Ref. [13], at least the first order factorizable terms would be gained analytically, leaving the non-factorizable part behind for further analysis using different methods, like those for elliptic integrals and their potential generalization.

Having the above number of moments, Eq. (1) yields a first numerical approximation of $\hat{J}_k(x)$ also up to the required power in ε .

The present method is suitable to obtain precise numerical representations in case of various current massless and massive calculations at 3- and 4-loop order. In case one may solve the associated difference equations algebraically, one will even obtain the complete analytic result without any prejudice, i.e. a sufficiently large set of scalar moments allows one to fully reconstruct a one-dimensional distribution. This automatic method is therefore suited to carry out many more higher loop calculations in contemporary elementary particle physics in a completely or at least widely analytic form.

We would like to thank J. Ablinger, A. Behring, A. De Freitas, M. Kauers, and P. Marquard for discussions. This work was supported in part by the Austrian Science Fund (FWF) grant SFB F50 (F5009-N15), the European Commission through contract PITN-GA-2012-316704 (HIGGSTOOLS).

[1] J.A.M. Vermaseren, A. Vogt and S. Moch, Nucl. Phys. B **724** (2005) 3 [hep-ph/0504242].
 [2] B. Ruijl et al. PoS (LL2016) 071 [arXiv:1605.08408 [hep-ph]].
 [3] I. Bierenbaum, J. Blümlein and S. Klein, Nucl. Phys. B **820**, 417 (2009) [arXiv:0904.3563 [hep-ph]].

[4] J. Blümlein, D.J. Broadhurst and J.A.M. Vermaseren, Comput. Phys. Commun. **181** (2010) 582 [arXiv:0907.2557 [math-ph]];
 J. Ablinger, J. Blümlein and C. Schneider, J. Phys. Conf. Ser. **523** (2014) 012060 [arXiv:1310.5645 [math-ph]].
 [5] J. Ablinger et al., DESY 16-147.
 [6] Standard algorithms like MINCER and MATAD [7] do only allow the calculation of a comparably low number of moments of up to $O(20 - 30)$ at 3-loop order.
 [7] S.A. Larin, F.V. Tkachov and J.A.M. Vermaseren, NIKHEF-H-91-18; S.G. Gorishnii et al., Comput. Phys. Commun. **55** (1989) 381;
 M. Steinhauser, Comput. Phys. Commun. **134** (2001) 335 [hep-ph/0009029].
 [8] M. Kauers, *Guessing Handbook*, Version 0.32, 2009-04-07, RISC, Linz.
 [9] J. Blümlein et al., Comput. Phys. Commun. **180** (2009) 2143 [arXiv:0902.4091 [hep-ph]].
 [10] 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, 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]];
 [11] V.N. Velizhanin, arXiv:1411.1331 [hep-ph].
 [12] K.G. Chetyrkin and F.V. Tkachov, Nucl. Phys. B **192** (1981) 159.
 [13] J. Ablinger et al., Comput. Phys. Commun. **202** (2016) 33 [arXiv:1509.08324 [hep-ph]].
 [14] A. Danilevskiĭ, Mat. Sbornik **2** (1937) 169;
 M.A. Barkatou, Appl. Algebra Engrg. Comm. Comput. **4**(3) (1993) 185;
 B. Zürcher, *Rationale Normalformen von pseudo-linearen Abbildungen*, Master's thesis, ETH Zürich (1994).
 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 2013, Boston, arXiv:1301.5414 [cs.SC] and references therein.
 [15] S. Gerhold, *Uncoupling systems of linear Ore operator equations*, Master's thesis, RISC, J. Kepler University, Linz, 2002.
 [16] J. Blümlein, S. Klein, C. Schneider and F. Stan, J. Symbolic Comput. **47** (2012) 1267 [arXiv:1011.2656 [cs.SC]].
 [17] C. Schneider, A. De Freitas and J. Blümlein, PoS (LL2014) 017 [arXiv:1407.2537 [cs.SC]].
 [18] J. Ablinger et al., PoS (RADCOR2015) 060, arXiv:1601.01856 [cs.SC].
 [19] J. Ablinger et al., PoS (LL2016) 005, arXiv:1608.05376 [cs.SC].
 [20] J. Ablinger et al., Nucl. Phys. B **885** (2014) 280 [arXiv:1405.4259 [hep-ph]].
 [21] J.A.M. Vermaseren, Int. J. Mod. Phys. A **14** (1999) 2037 [hep-ph/9806280];
 J. Blümlein and S. Kurth, Phys. Rev. D **60** (1999) 014018 [hep-ph/9810241].
 [22] J. Ablinger et al., J. Math. Phys. **55** (2014) 112301 [arXiv:1407.1822 [hep-th]].
 [23] J. Ablinger et al., in preparation.