

MB tools reloaded

A.V. Belitsky^{a,*}, A.V. Smirnov^b, V.A. Smirnov^c

^a Department of Physics, Arizona State University, Tempe, AZ 85287-1504, USA

^b Research Computing Center, Moscow State University and Moscow Center for Fundamental and Applied Mathematics, 119992 Moscow, Russia

^c Skobeltsyn Institute of Nuclear Physics, Moscow State University and Moscow Center for Fundamental and Applied Mathematics, 119992 Moscow, Russia

Received 9 December 2022; accepted 20 December 2022

Available online 28 December 2022

Editor: Stephan Stieberger

Abstract

We address the problem of evaluation of multiloop Feynman integrals by means of their Mellin-Barnes representation. After a brief overview of available capabilities through open source toolkits and their application in various circumstances, we introduce a new code `MBcreate` which allows one to automatically deduce a concise Mellin-Barnes representation for a given parametric integral. A thorough discussion of its implementation and use is provided.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>). Funded by SCOAP³.

1. Overview

Feynman parametrization of momentum (or position, for that matter) space integrals is undoubtedly the most widespread tool to perform D -dimensional loop integrals – see, e.g., recent books [1,2]. After this rather straightforward step, one ends up with an N -fold parametric integral of the form

$$I_N(\{s\}) = \int_0^\infty d^N x \, \delta\left(\sum_{i=1}^N x_i - 1\right) \mathcal{J}(\{x\}; \{s\}), \quad (1)$$

* Corresponding author.

E-mail address: andrei.belitsky@asu.edu (A.V. Belitsky).

where the integrand

$$\mathcal{J}(\{x\}; \{s\}) = \mathcal{U}^{p_1}(\{x\}) \mathcal{F}^{p_2}(\{x\}; \{s\}) \prod_{i=1}^N x_i^{a_i-1}, \quad (2)$$

is encoded in two Symanzik functions $\mathcal{U}(\{x\})$ and $\mathcal{F}(\{x\}; \{s\})$, depending on Feynman parameters $\{x\} = \{x_1, \dots, x_N\}$ and kinematical invariants $\{s\} = \{s_1, \dots, s_M\}$. The former are defined by trees and 2-trees of a given Feynman graph, respectively. $\mathcal{U}$ is a linear in each x_i polynomial with positive coefficients. The dependence on external kinematical invariants and masses s_k enters linearly through the $\mathcal{F}$ polynomial only. In the case of vanishing masses, the latter is linear in each x_i as well but not otherwise. These graph polynomials possess, correspondingly, the degrees of homogeneity h and $h+1$ in Feynman parameters, $\mathcal{U}(\{\lambda x\}) = \lambda^h \mathcal{U}(\{x\})$ and $\mathcal{F}(\{\lambda x\}; \{s\}) = \lambda^{h+1} \mathcal{F}(\{x\}; \{s\})$, where h is the number of loops. Finally, the exponents of these polynomials are $p_1 = a - (h+1)D/2$ and $p_2 = -a + hD/2$ with $a = \sum_{i=1}^N a_i$ and individual a_i 's corresponding to the powers of appearing propagators in the initial momentum integrand. An extensive discussion of their construction can be found in Refs. [3,1].

The focus of the present paper is on the calculation of (1) by transforming it from the real axis to the complex plane where information about integrand's singularities will be sufficient to compute I_N making use of powerful theorems of the Complex Analysis. The starting point for this well-known method is based on the following Mellin-Barnes (MB) representation

$$\frac{1}{(A+B)^\lambda} = \frac{1}{\Gamma(\lambda)} \int_{\mathcal{C}} \frac{dz}{2\pi i} \frac{B^z}{A^{\lambda+z}} \Gamma(\lambda+z) \Gamma(-z), \quad (3)$$

which allows one to partition a complicated polynomial in terms of its two 'simpler' components A and B . In this equation, the contour $\mathcal{C}$ goes from $-i\infty$ to $+i\infty$ in the complex plane and the poles of $\Gamma(\dots+z)$ are to its left while the ones of $\Gamma(\dots-z)$ are to its right with these left/right poles corresponding to infrared/ultraviolet singularities of the original integral. This formula is usually applied repeatedly enough number of times to a given parametric integral I_N in order to solve all x -integrations in terms of products of Euler Gamma functions. This yields a sought-after MB representation for a given Feynman integral in the form of an n -fold complex integral (generally $n \neq N$)

$$\int_{\mathcal{C}_1} \dots \int_{\mathcal{C}_n} \frac{d^n z}{(2\pi i)^n} \frac{\prod_i \Gamma(\alpha_i + \beta_i \epsilon + \sum_j \gamma_{ij} z_j)}{\prod_i \Gamma(\alpha'_i + \beta'_i \epsilon + \sum_j \gamma'_{ij} z_j)} \prod_k s_k^{d_k}. \quad (4)$$

The 'additive' dependence of the second Symanzik polynomial $\mathcal{F}(\{x\}; \{s\})$ on the kinematical invariants/masses s_k is thus transformed into the multiplicative dependence on their d_k -powers.

The above MB representation (4) was successfully employed in analytical calculations of Feynman integrals starting with the seminal work of Refs. [4,5] where three- and four-point massless ladders at generic values of squared external momenta were obtained. Since multiloop Feynman integrals are rather involved objects, they are usually evaluated as a Laurent expansion in the parameter of dimensional regularization $\epsilon = (4-D)/2$, rather than for generic D values. Emerging poles in ϵ have different origin reflecting divergent regions in the initial momentum integrals: they can stem from ultraviolet, infrared, collinear etc. domains. A systematic study of MB representation (4) for dimensionally regularized Feynman integrals was initiated in Refs. [6,7] where two complementary strategies for resolving occurring singularities in ϵ near

$\epsilon = 0$ were devised. More than that, two public computer codes based on these techniques were developed in [8,9], respectively. This laid out the foundation for a widespread use of the MB techniques by QFT practitioners, see, e.g., Chapter 5 of [1] for a review.

Admittedly the MB method had seen its better days in the rear-view mirror as it passed its pinnacle on the stage of calculation tools being superseded by the introduction of canonical integral bases [10] within the method of differential equations (DEs) [11,12]. A historical remark is in order to make this point clear to the reader. To reveal the so-called BDS Ansatz [13] for four-gluon scattering amplitudes at three-loop order, it was necessary to evaluate two four-leg Feynman integrals associated with triple ladder-box and tennis-court graphs. This was performed in [14] and [13], respectively, making use of the MB technique at the time when no computer codes were yet available. The resolution of integrand's singularities was the bottleneck of its successful calculation, making the task of its manual examination colossally tedious. With availability of the codes [8,9], this complication can be immediately alleviated. Still application of the same approach to all master integrals of the above two families of Feynman integrals would not be even feasible, but with the use of DEs for canonical bases of integrals this goal can successfully be achieved as was shown in Ref. [15]. More than that, in a similar manner, master integrals for all four-leg massless on-shell non-planar graphs were also evaluated [16,17].

Nevertheless, the MB method remains powerful enough to keep its runner-up position and can be applied in conjunction with DEs in order to fix their boundary conditions, see, e.g., Refs. [18,19], or, in certain circumstances, it is the only available choice when DEs cannot be used or face their own vices. A particularly suitable niche for the application of the MB technique is in the analysis of asymptotic behavior of Feynman graphs for small/large values of occurring kinematical invariants/masses s_k : like Sudakov and heavy mass limits, just to name a few. Leading contributions in these cases are revealed with the help of a strategy known as the Expansion by Regions [20] (see also [21,22,1]). This is accomplished by applying the public Mathematica code `asy` [23,24], – also available as the `SDExpandAsy` command with the FIESTA5 distribution package [25], – which is based on the analysis of the geometry of polytopes associated with the two Symanzik polynomials $\mathcal{U}(\{x\})$ and $\mathcal{F}(\{x\}; \{s\})$. It determines all leading contributions to the I_N integral by scanning over various scaling behaviors of the Feynman parameters with asymptotic values of kinematical invariants. The output is given as parametric integrals of the I_N type but with reduced, scale-independent Symanzik polynomials, $\tilde{\mathcal{U}}(\{x\})$ and $\tilde{\mathcal{F}}(\{x\})$. Since there is no dependence on kinematical variables left, DEs are powerless and the MB approach is the only game in town. This strategy was recently applied on different occasions, see [26,27], which compelled updates to existing routines of the MB toolbox as well as development of a new code, which will be described below.

The subsequent presentation is organized as follows. Sect. 2 describes the main contribution of this work through the code `MBcreate`, which generates a concise MB representation for a given Feynman integral. Next, Sect. 3 provides an exposition of existing codes connected with the MB representations, which allows one to solve MB integrals in the form of the Laurent expansion in ϵ with analytic coefficients expressed in terms of Riemann zeta values. Conclusions with several appendices culminate the paper.

2. Introducing `MBcreate.m`

The first of order of business on the way to apply available MB tools is to derive an optimal MB representation (4) for a given Feynman-parameter integral (1) with a minimal number n_0 of complex integrations. For generic momentum-space integrals, one can proceed in two different

ways: construct Feynman parametric representation for the entire multiloop integrand, then deduce corresponding MB integrals, or do it loop-by-loop, i.e., derive an MB representation for a one-loop subintegral, then embed it into a larger two-loop integral and so on. It turns out that the global route does not yield the minimal value for n_0 . An example to this point is the very first analytical calculation of dimensionally regularized double boxes [6] where the global parametric representation produced five MB integrations but later it was observed [28] that in the loop-by-loop approach that it reduces down to four. The latter method was then successfully used in planar-graph calculations, e.g., [14,13]. It found its automatic implementation in the public code `AMBRE.m` [29]. Yet another algorithm to derive an optimal MB representation was proposed in Ref. [30], however, its computer implementations is not available so far.

Starting with a generic Feynman integral (1) depending on M kinematical variables s_k , it is sometimes natural to isolate them first in a factorized form (4) by means of the repeated use of Eq. (3) at the cost of introducing $M - 1$ MB integrations. This can be done with the option `SplitExtraVars -> True`. However, it is usually better not to do this so that the default option is `SplitExtraVars -> False`. Anyway, an input for our code is a product of several parametric polynomials F_j (with positive coefficients) raised to certain (generally) complex powers. Similarly, as discussed in the previous section, the application of expansion by regions yields s_k -independent reduced Symanzik polynomials in parametric integrands. In either case, one has to construct an optimal MB representation for these. One can of course proceed by trial and error on a case-by-case basis looking for the magic number n_0 . This was done in Refs. [26,27]. However, this is extremely time-consuming. In the lack of a proof of what a numerical value n_0 might be a priory, a routine that can search for its optimal value needs to be developed to tackle this problem. So the lowest value of n_0 that it finds will constitute an efficiency criterion.

Therefore, consider a Feynman parametric integral independent of kinematical invariants

$$\int_0^\infty d^N x \prod_i x_i^{a_i-1} \delta\left(\sum_{i=1}^N x_i - 1\right) \prod_j F_j^{p_j}(\{x\}) . \quad (5)$$

Here F_j are polynomials with positive coefficients linear in each x_i , raised to powers $p_j = b_j \epsilon + c_j$; a_j are integers, while b_j, c_j are rational numbers when these are thought of as outputs of the Expansion by Regions,¹ or complex when it is a result of kinematical split-up alluded to at the top of the previous paragraph. Notice that a_i 's can also be considered generally complex-valued if an auxiliary analytic regularization is imposed. This latter setup is particularly relevant for initially finite parametric integrals where one can choose to set the number of space-time dimensions down to four, i.e., $\epsilon = 0$. However, since its asymptotic expansion with expansion by regions generates individually divergent contributions an intermediate regularization is nevertheless required. It has to be imposed however in a manner that does not violate the rescaling invariance of the original parametric integral under $\{x\} \rightarrow \{\lambda x\}$ transformation. The preservation of this property is crucial for maintaining the opportunity to apply the so-called Cheng-Wu theorem [31] to the above integral. For reader's convenience and completeness of this presentation, the theorem is reviewed in Appendix A and boils down to reducing the delta-function constraint down to a smaller subset of Feynman parameters. A particularly convenient choice is $\delta(x_{i_0} - 1)$ for a single ad hoc i_0 . In certain calculations, one eliminates the delta function constraint first in favor of symmetric treatment of all integrals involved, be it Feynman-parameter or proper-time

¹ In this case, it is obvious that $F_1 = \tilde{\mathcal{U}}$ and $F_2 = \tilde{\mathcal{F}}$.

integrals, see, e.g., [26]. The procedure devised below is applicable to those circumstances as well.

The procedure is built on the following two transformations: integration over an x -parameter, if possible, making use of the integral

$$\int_0^\infty dx x^p (ax + b)^q = \frac{\Gamma(p+1)\Gamma(-p-q-1)}{\Gamma(-q)} a^{-p-1} b^{p+q+1}, \quad (6)$$

introduction of an MB integration (3) in order to apply (6). Obviously, the first x -integrations which have to be performed in Eq. (5) are the ones over non-overlapping subsets of variables defining the F_j polynomials. Without loss of generality it suffices to address the case of just two polynomials in the integrand F_1 and F_2 (see footnote 1). Suppose that there are several variables with this property, i.e., F_2 depends on all of the Feynman parameters while F_1 is independent of a subset $\Sigma = \{x_\ell\}$ of these.² Then $F_2 = F_{2,1}x_\ell + F_{2,0}$ and by means of (6), we obtain the product $F_{j,1}^{p_1} F_{j,0}^{p_0}$. Then one repeats this step for the next variable from Σ provided it belongs to either $F_{j,1}^{p_1}$ or $F_{j,0}^{p_0}$ but not both. After such integrations become impossible, one is forced to use the MB partition (3) first before applying (6) again. It is at this step that an optimal choice of the decomposition of the progenitor polynomial $F_{j,\dots}$ into its simpler components becomes crucial for the most efficient MB representation. The key question is to minimize the number n_0 of those complex integrations.

The generic steps outlined above were implemented in the Mathematica package `MBcreate.m`, which attempts to minimize the value of n_0 . In particular, `MBcreate.m` examines and applies the following procedures one-by-one, not necessarily in the order listed, unless it is explicitly specified.

- Factorizes kinematic invariants s_k from F_j 's: if $F_j = f_{j,0} + s_k f_{j,1}$, an MB representation (3) is introduced to split up $f_{j,0}$ and $s_k f_{j,1}$.
- Implements the change of variables $x_i = \eta\xi$, $x_j = \eta(1 - \xi)$ for two Feynman parameters entering integrands and obeying the conditions: (i) the dependence of each of the functions F_j on η is at most linear, (ii) no more than two F_j 's depend on it. Otherwise, introduces an MB representation, integrates with respect to η . Next introduces yet another MB decomposition (3) with subsequent integration over ξ .
- Searches through all $\{x_i, x_j\}$ pairs and find cases where only one of the F_j function depends on a single variable, say x_i , not the sum of the two $x_i + x_j$. Splits up that function into two terms, one depending on the sum and the rest, solves the resulting integration with Eq. (6).
- Tries all decompositions of the form $F_j = x_i F_{j,1} + F_{j,0}$, where $F_{j,0}$ does not depend on x_i , with both $F_{j,1}$ and $F_{j,0}$ being factored into monomials accompanying residual polynomials. Splits F_j by introducing an MB integration.
- Scans all decompositions $F_j = x_i F_{j,1} + F_{j,0}$, where $F_{j,0}$ is x_i -independent and splits them up with the MB representation (3) provided $F_{j,1}$ and/or $F_{j,0}$ already exist in the list of functions populating the integrand. This reduces the number of polynomials which could potentially yield a higher value of n_0 .
- Searches for possible splitting based on the form $F_j = x_i F_{j,1} + F_i$ where F_i is one of the factors already present in the integrand.

² This is a typical situation for a bulk of contributions stemming from expansion by regions.

- Tries the decompositions $F_j = x_i F_{j,1} + F_{j,0}$ where $F_{j,0}$ can depend on x_i but is factorized into a product of simpler, lower-degree polynomials.
- Searches for ‘similar’ functions F_i and F_j defined by the condition that $F_i - F_j$ is given by a difference of two monomials. Splits one of them, say F_i , into the monomial associated with it and the rest, even more cognate with F_j function, by means of Eq. (3).
- If none of the above procedures meet their requirements, chooses a Feynman variable x_i , introduces MB representations for all F_j ’s in the integrand but one and performs the integration over x_i of the last remaining polynomial with the help of Eq. (6).

`MBcreate.m` automatically applies all of the above strategies, prioritizing the search for $\{x_i, x_j\}$ pairs, and then solves the resulting parameter integrals whether they require an MB representation or not. The output is given by the product of ratios of Euler Gamma functions as in the integrand of Eq. (4). Though, currently there is no proof that the achieved value of n_0 is minimal, authors’ experience and multiple tests against “manual reductions” give sufficient confidence to expect that.

The splitting of kinematical variables can be turned on/off with an option `SplitExtraVars` $\rightarrow$ `True/False` and is explicitly demonstrated in Section 4.

On extremely rare occasions, when an additional regularization is called for successful resolution of singularities discussed in the next section, an output can be encountered with Euler Gamma of arguments depending on the parameter of analytic regularization only. These have to be scrapped and redone by manually reshuffling the indices of x ’s.

3. MB tools overhauled

Having derived an MB representation (4) for a parametric integral, one has to solve it either exactly or as a Laurent series in ϵ up to a desired order, with coefficients which are given by MB integrals independent of ϵ , i.e., pure numbers. This section discusses a general strategy of accomplishing this goal. It is demonstrated using specific example of Feynman integrals in the next section.

First, it is necessary to resolve the singularity structure in ϵ . As was already addressed in the introductory section, one can use either `MB.m` or `MBresolve.m` for that purpose, which were delivered in Refs. [9,8], respectively. The initial point of `MB.m` is to apply `MBoptimizeRules` command in order to find straight contours and values of ϵ obeying the rules for the contour choice formulated immediately after Eq. (3). Such contours do not always exist from the get-go. To alleviate the problem one can introduce an auxiliary analytic regularization complementary to the dimensional one and then proceed to contour determination with this command. There is no universal prescriptions how to do this in a systematic way and it is not straightforward. Due to these complications, it is preferable to rely on `MBresolve.m` instead. As it was explained in detail in Ref. [8], the code searches for optimal straight contours for the resolution of singularities in ϵ . Only on rare occasions, the code is unable to perform and this calls for an auxiliary analytic regularization. A recommended way of doing it in a systematic fashion is to provide additive terms to all a_i in Eq. (5) proportional to a parameter, say, λ , i.e., $a_i \rightarrow a_i + r_i \lambda$ with the total sum $\sum_i r_i$ equal to zero.³ The second reason why it is advantageous

³ The last condition is important because it does not affect re-parametrization invariance of parametric integrands to choose a “gauge” condition on one of the x ’s. Also numerical checks of Laurent expansions with `SDExpandAsy` command of `FIESTA` can be used provided this condition is fulfilled.

to apply `MBresolve.m` rather than `MB.m` is that `MBresolve.m` is much faster and this turns out rather important if the number of MB integrations is very large. A comment is in order about these two codes: every so often they used to produce real shifts of the contour which were integers this yielded error messages in the subsequent steps of analyses. The matter was resolved by increasing the precision⁴ of the conversion of decimal to rational output to 10^{-5} .

The next step is to evaluate pure number MB integrals involved. This is accomplished by running the command `DoAllBarnes` from `barnesroutines.m` [32] which automatically applies the first and the second Barnes lemmas and thereby performs some integrations in terms of Euler Gamma functions. The current version of the routine does not include a plethora of corollaries of the lemmas and they have to be applied by hand as in recent studies [27]. A case in point is

$$\begin{aligned} \int_C \frac{dz}{2\pi i} \frac{\Gamma(a+z)\Gamma(-b-z)\Gamma(b+z)\Gamma(d-z)}{z} &= -\Gamma(2-a)\Gamma(a)\Gamma(-b)\Gamma(b) \\ &+ \Gamma(2-a)\Gamma(-b)\Gamma(a-b-1)\Gamma(b-a+2) \frac{\psi(1-a) - \psi(-b)}{\Gamma(1-a)\Gamma(1-b)} \\ &- \frac{1}{b^2} \Gamma(a-b)\Gamma(b-a+2)(b(\psi(b-a+2) + \gamma_E) - 1), \end{aligned} \quad (7)$$

where the pole $z=0$ stays to the left of an integration contour and the pole $z=-b$ positioned to the right of the integration contour. A very long comprehensive list of similar formulas is provided in the Appendix B as an attachment. Let us also mention various ways [33] to integrate explicitly subintegrals in MB integrals which might be also implemented in `DoAllBarnes`.

Since the use of these tables does not warrant successful calculation of integrals, one has to rely on alternative techniques such as experimental mathematics. Namely, a much faster route to find analytical expressions to MB integrals in many circumstances, however, is immediately after the application of `DoAllBarnes` to bypass the use of corollaries of Barnes lemmas and turn to numerical analyses of remaining MBs. In practice, the computation of the latter is not problematic since these converge very well at large imaginary values of z -integration variables due of the exponential suppression stemming from Euler Gamma functions involved. Thus, calculating these with sufficiently high precision is possible. Then one can use the PSLQ algorithm [34] to obtain analytic results provided a basis of numbers, typically values of Riemann zeta function, entering the final result is known. For one-dimensional MB integrals, the current version of `NIntegrate` with `GlobalAdaptive` strategy in `Mathematica` can achieve the precision of 100 or more with ease and then the built-in command `FindIntegerNullVector` allows one to successfully recognize transcendentals.⁵ Currently, `Mathematica` cannot handle well n -fold integrals for $n > 2$ with sufficiently high precision and this sets a strong limitation of this calculational strategy. For instance, for two-fold MB integrals, only an older version of `Mathematica`, e.g., v.5.2, permits one to gain sufficient precision (topping at 40) with the `DoubleExponential` option for `NIntegrate`. A lower available precision imposes a very strong restriction on the dimension of the basis of transcendental numbers. After analytic expressions have been found with PSLQ, it is advisable to use `MBintegrate` of `FIESTA` to the intermediate output of `MBresolve` to verify the former numerically. At the time of writing this

⁴ If this obstruction still persists, a user can further increase the precision by opening the packages `MB.m` and `MBresolve.m`, searching for `Rationalize` command and making the change by her/himself.

⁵ A code implementing it is provided in Appendix C.

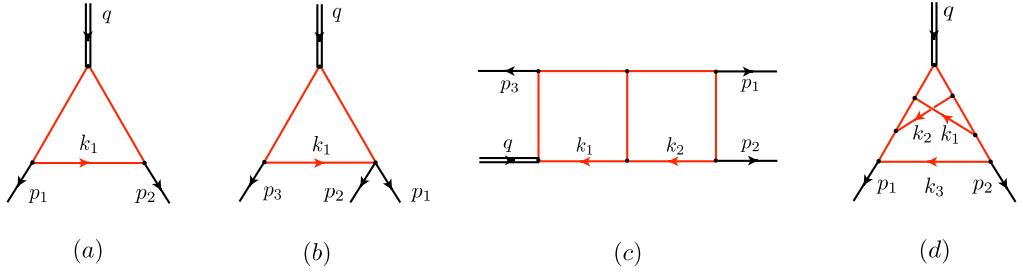


Fig. 1. Exemplary graphs analyzed with MB Tools in the notebook MBcreateUsage.nb attached with submission.

article, for MBs with $n = 3$ and higher, one has to be content with possessing numerical results only or come up with a different, more powerful technique.

While the original MB package was distributed via [hepforge](https://mbtools.hepforge.org/) <https://mbtools.hepforge.org/>, the current development of MBcreate.m is undergone with the use of `git` and `bitbucket`. Both MBcreate.m and all other MB codes are collected in the same repository and can be freely downloaded from there: <https://bitbucket.org/feynmanIntegrals/mb/src/master/>.

Most of the codes require `Mathematica` and simply work when loaded there. However, the MBintegrate command performs integration with the use of `fortran` generated codes and requires the `gfortran` compiler (which can normally be installed with package managers such as `apt-get`) as well as some libraries. While the Cuba integration library [35] by T. Hahn is shipped with the package and works perfectly with modern compilers, the original MB code used also the `cernlib` library for the evaluation of polylogarithms, but the `cernlib` is no longer supported, and there might be a problem to install it at modern computers. Hence a code based on a small portion of `cernlib` which was provided by M. Czakon is also included in the repository. All libraries can be compiled by calling `make` in the package folder.

The new package is also accompanying this submission as an ancillary file, for reader's convenience, along with a `Mathematica` notebook MBcreateUsage.nb with thoroughly worked out examples as discussed in the next section.

4. Examples

This section demonstrates the use of the code MBcreate.m applying it to a few examples of increasing complexity along with other routines from the MB Toolbox.

4.1. Graph 1(a)

The essence of the code can be shown with a one-loop diagram displayed in Fig. 1 (a). Using FIESTA [25] to generate corresponding Symanzik polynomials U and F ,

```
In[1] := props1L = {-k1^2, -(k1 + p1)^2, -(k1 - p2)^2};
        reps1L = {p1^2 -> -S1, p2^2 -> -S2, p1*p2 -> (S1 + S2 - S12)/2};
        IniFI = UF[{k1}], props1L, reps1L]
Out[1] :=
{x[1] + x[2] + x[3], S1 x[1] x[2] + S2 x[1] x[3] + S12 x[2] x[3], 1}
```

one can feed them in directly to the `MBcreate`. Depending on whether one instructs the code to split up external kinematical variables at the first step or not with the help of the option `SplitExtraVars`, one gets equivalent representation for the graph 1 (a)

```
In[1]:= MBcreate[IniFI, Verbose -> True, SplitExtraVars -> True]
Out[1]:= (S1^z[1] S12^z[2] S2^(-1 - ep - z[1] - z[2]) Gamma[-ep - z[1]]
          Gamma[-z[1]] Gamma[-ep - z[2]] Gamma[-z[2]] Gamma[1 + z[1] + z[2]]
          Gamma[1 + ep + z[1] + z[2]])/Gamma[1 - 2 ep]
```

and

```
In[1]:= MBcreate[IniFI, Verbose -> True, SplitExtraVars -> False]
Out[1]:= (1/Gamma[1 - 2 ep]) S1^(-1 + ep - z[1]) S12^(-2 ep + z[1] - z[2])
          S2^z[2] Gamma[1 - 2 ep + z[1]] Gamma[1 - ep + z[1]] Gamma[-ep - z[2]]
          Gamma[-z[2]] Gamma[ep - z[1] + z[2]] Gamma[2 ep - z[1] + z[2]]
```

respectively.

4.2. Graph 1(b)

Having established what `MBcreate` does, one may use it as a part of an effort to calculate Feynman graphs as function of the kinematical invariants and ϵ parameter of dimensional regularization. A case in point is the diagram in Fig. 1 (b). Considering it, for instance, in the limit of small virtuality $p_3^2 \equiv t p_{123}^2$ as $t \rightarrow 0$, but exactly in the other two variables⁶ p_{12}^2 , p_{123}^2 and ϵ . The asymptotic expansion of the graph is achieved with the help of the Method of Regions [20] making use of the FIESTA command `SDExpandAsy`. Then we immediately get two contributing domains

```
In[1]:= props1L = {-k1^2, -(k1 + p3)^2, -(k1 - p12)^2};
reps1L = {p3^2 -> -t S123, p12^2 -> -S12,
p3*p12 -> (t S123 - S123 + S12)/2};
SDExpandAsy[UF[{k1}], props1L, reps1L], {1, 1, 1}, {0, 0, 0},
AnalyticIntegration -> False, UsingC -> False,
OnlyPrepareRegions -> True, QHullPath -> dir]
Out[1]:= {{{{0, 0, 0}, {E^(ep EulerGamma) Gamma[1 + ep]
FIESTA`Private`KronekerDelta[x[1] + x[2] + x[3], 1]
(x[1] + x[2] + x[3])^(-1 + 2 ep)
(S12 x[1] x[3] + S123 x[2] x[3])^(-1 - ep)}},
{{0, 0, 1}, {E^(ep EulerGamma) t^-ep Gamma[1 + ep]
FIESTA`Private`KronekerDelta[x[1] + x[2] + x[3], 1]
(x[1] + x[2])^(-1 + 2 ep)
(S123 x[1] x[2] + S12 x[1] x[3] + S123 x[2] x[3])^(-1 - ep)}}}}
```

Starting from the first region $\{0, 0, 0\}$, it is fed into `MBcreate` to generate the sought after MB representation

```
In[1]:= Inil = {1, x[1]^la[1] x[2]^la[2] x[3]^(-1 - ep + la[3]),
{{S12 x[1] + S123 x[2], -1 - ep}, {x[1] + x[2] + x[3], -1 + 2 ep}}},
```

⁶ Here and below, we use the shorthand notation $p_{ijk\dots} = p_i + p_j + p_k + \dots$

```

False}
MBcreate[In1*Product[x[i]^la[i], {i, 1, 3}], Verbose -> True,
SplitExtraVars -> True]
Out[1]:= (S12^z[1] S123^(-1 - ep - z[1]) Gamma[1 - ep - la[3]]
Gamma[-ep + la[3]] Gamma[-ep + la[2] - z[1]] Gamma[-z[1]]
Gamma[1 + ep + z[1]] Gamma[1 + la[1] + z[1]])
/(Gamma[1 - 2 ep] Gamma[1 + ep] Gamma[1 - ep + la[1] + la[2]])

```

Notice that an additional factor $x[1]^{la[1]} x[2]^{la[2]} x[3]^{la[3]}$ is introduced to the second, monomial element of the input array. Its sole purpose is to provide an additional analytic regularization for safe use of subsequent codes from the MB Toolbox.

The next goal is to calculate the arising MB integral analytically. This can be achieved by first constructing its proper straight integration contour as one moves the original curved contour separating positive and negative poles of Gamma functions and resolving arising singularities. This is automated by the command `MBresolve`, where `MBintegral1` is the output of the `MBcreate`

```

In[1]:= MBintegralCont = MBresolve[MBintegral1, ep, OptimizeNow -> True]
Out[1]:= {MBint[(S12^z1 S123^(-1 - ep - z1) Gamma[-ep] Gamma[-ep - z1] Gamma[-z1]
Gamma[1 + z1] Gamma[1 + ep + z1])/(Gamma[1 - 2 ep] Gamma[1 + ep]),
{ep -> 0}, {z1 -> -(5935/11869)}]}

```

Since `MBresolve` worked out without a hiccup for zero values of la , so they were ignored from the get-go. It is advisable at this time to have a numerical value for the integral for some random value of parameters for verification of future analytical result with `MBintegrate`.

Having done that, the single-fold MB integral can be recast as an infinite sum and then evaluated by *Mathematica*. The former is accomplished with the package `MBsums` [36], by running

```

In[1]:= Lk = {S12 -> 1/2, S123 -> 1/3};
dimlsum = MBIntToSum[MBintegralCont[[1]], Lk, {z1 -> L}]
Out[1]:= {MBsum[((-1)^-n1 S12^n1 S123^(-1 - ep - n1) Gamma[-ep]
Gamma[-ep - n1] Gamma[1 + ep + n1])/(Gamma[1 - 2 ep] Gamma[1 + ep]),
n1 >= 0, {n1}],
MBsum[((-1)^-n1 S12^(-ep + n1) S123^(-1 - n1) Gamma[-ep]
Gamma[ep - n1] Gamma[1 - ep + n1])/(Gamma[1 - 2 ep] Gamma[1 + ep]),
n1 >= 0, {n1}]}

```

Then, *Mathematica*'s built-in `Sum` yields the final answer for the first region

```

Region1Res = (S123^-ep Gamma[-ep]^2)/((-S12 + S123) Gamma[1 - 2 ep])
+ (S12^-ep Gamma[1 - ep] Gamma[-ep] Gamma[ep])/((-S12 + S123)
Gamma[1 - 2 ep] Gamma[1 + ep]);

```

Completely analogous steps hold for the second region $\{0, 0, 1\}$. The reader is spared however this unnecessary reiteration and instead is instructed to consult the accompanying notebook `MBcreateUsage.nb` for details. The same cell also contains information on the Laurent expansion of the total result in ϵ with subsequent simplification of the output making use of the symbol technology [37].

4.3. Graphs 1(c) and (d)

Application of `MBcreate` does not face any complications at higher loop orders, like graphs (c) and (d) in Fig. 1. Due to the space constraint, all of the details are relegated to the accompanying notebook of examples. However, it is important to point out that what changes drastically at higher loops is one's ability to evaluate MB integrals analytically. In those circumstances, when *Mathematica* fails to evaluate multiple sums obtained from a given multifold MB integral, the best hope one can have is to rely on experimental mathematics implemented in the PSLQ algorithm alluded to in Section 3 with explicit implementation in Appendix C. In particular, it is indispensable for analytic reconstruction of the asymptotic expansion of the three-loop graph in Fig. 1 (d).

5. Conclusion

This work introduced a new package for the conversion of Feynman integrals into an MB form with a minimal number of complex integrations, `MBcreate`. Also an update to several routines in the MB toolbox was provided to have error-free outputs at each step of analytical calculation of Laurent expansion of Feynman integrals. These were thoroughly tested against calculations done mostly “by hand” in Refs. [26,27].

For completeness, it is worth pointing out that while the strategy outlined above heavily relies on the PSLQ algorithm, there is yet another alternative way to evaluate MB integrals by transforming them into infinite series representation by closing integration contours and taking residues with a help of computer code presented in Ref. [36]. The very problem of finding series representations for a given MB integrals was recently analyzed in [38,39] making use of an approach based on conic hulls. A public computer code was also given there.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

We would like to thank M. Czakon for providing the codes required to make `MBintegrate` to work on modern computers. The work of A.B. was supported by the U.S. National Science Foundation under the grant No. PHY-2207138. The work of A.S. and V.S. was supported by the Russian Science Foundation under the agreement no. 21-71-30003 (application of MB representations to research in elementary particle physics) and by the Ministry of Education and Science of the Russian Federation as part of the program of the Moscow Center for Fundamental and Applied Mathematics under Agreement No. 075-15-2019-1621 (development of algorithms creating MB representations).

Appendix A. Folklore Cheng-Wu theorem

Consider an N -fold integral on the standard simplex

$$I_N = \int_{\mathcal{S}_x} d^N x J(\{x\}), \quad \mathcal{S}_x = \left\{ \{x\} \in \mathbb{R}^N : \sum_{i=1}^N x_i = 1, x_i \geq 0 \right\}, \quad (8)$$

with the integrand being a homogeneous function of the x_i variables of degree r , i.e., $F(\{\lambda x\}) = \lambda^r F(\{x\})$. This integral is not invariant under this rescaling instead it has the degree $r + N$. To alleviate this predicament, perform a projective transformation by passing to another set of variables $\{y\}$ as

$$x_i = y_i / \left(\sum_{i=1}^N y_i \right), \quad i = 1, \dots, N. \quad (9)$$

This change leaves the simplex domain invariant $\mathcal{S}_x = \mathcal{S}_y$. Then, taking into account the emerging Jacobian

$$d^N x = d^N y / \left(\sum_{i=1}^N y_i \right)^N, \quad (10)$$

the integral in these new variables becomes

$$I_N = \int_{\mathcal{S}_y} d^N y J(\{y\}) / \left(\sum_{i=1}^N y_i \right)^{r+N}, \quad (11)$$

and is explicitly rescaling invariant.

The latter property becomes crucial in the efficient solution of integrals by means of the application of the so-called Cheng-Wu theorem [31], which specifies various possible choices for multifold integrations. To formulate it, introduce a constraint on the integration variables in the integrand but integrate over unconstrained $\mathbb{R}_+^N$ space, i.e.,

$$I_N = \int_0^\infty d^N x \delta \left(\sum_{i=1}^N x_i - 1 \right) \mathcal{J}(\{x\}) \quad \text{with} \quad \mathcal{J}(\{x\}) = J(\{x\}) / \left(\sum_{i=1}^N x_i \right)^{r+N}. \quad (12)$$

The Cheng-Wu theorem states that one can freely change the argument of the Dirac delta function to

$$\sum_{i=1}^N x_i - 1 \rightarrow \sum_{i \in \Sigma} x_i - 1 \quad (13)$$

with Σ being a subset of N labels. In particular, one can choose just one, say i_0 , in which case this variable is set to $x_{i_0} = 1$ and the unconstrained integration is performed over the remaining $N - 1$ ones. It is important to realize that one could not have applied it to the original integral (8) since it is not rescaling invariant.

Though proofs of the Cheng-Wu theorem can be found in Refs. [40,1,2], it is enlightening however to present it again. One can directly prove it making use of the Stokes's theorem [2] applied to (12), but it is instructive to invoke instead its relation to the Feynman parameter integral for a graph as given in Eq. (2). Using it as a starting point, one can next integrate-in a variable to obtain the well-known Schwinger, aka alpha, representation for the integral

$$I_N = \frac{1}{\Gamma(a - hD/2)} \int_0^\infty d^N \alpha [\mathcal{U}(\{\alpha\})]^{-D/2} \exp\left(-\frac{\mathcal{F}(\{\alpha\})}{\mathcal{U}(\{\alpha\})}\right) \prod_{i=1}^N \alpha_i^{a_i-1}. \quad (14)$$

Obviously, the integrand (2) is obtained from this one by the rescaling $\{\alpha\} = \rho\{x\}$ and subsequent integration with respect to ρ . Using this representation as a starting point, the proof of the Cheng-Wu theorem becomes elementary. Namely, resolve the unity in terms of a constraint involving a subset of α 's, as on the right hand side of Eq. (13)

$$1 = \int_0^\infty d\sigma \delta\left(\sum_{i \in \Sigma} \alpha_i - \sigma\right), \quad (15)$$

and substitute it into the integrand of Eq. (14). Next, change the variables to $\{\alpha\} = \sigma\{x\}$ and use scaling properties of all functions involved to get the integral

$$I_N = \frac{1}{\Gamma(a - hD/2)} \int_0^\infty d^N x \prod_{i=1}^N x_i^{a_i-1} \delta\left(\sum_{i \in \Sigma} x_i - 1\right) \times [\mathcal{U}(\{x\})]^{-D/2} \int_0^\infty d\sigma \sigma^{a-hD/2-1} \exp\left(-\sigma \frac{\mathcal{F}(\{x\})}{\mathcal{U}(\{x\})}\right). \quad (16)$$

Finally integrating over σ gives original integrand with a constraint encompassing only a subset of integration variables.

Appendix B. Corollaries of Barnes lemmas (V.S. 2004)

For readers convenience, a very long list of corollaries of Barnes lemmas by V.S. is attached with this paper in the file `barnes.txt`.

Appendix C. PSLQ

Since the original Broadhurst's PSLQ code is not freely available to general public, here is a 'one-line' routine based on the `FindIntegerNullVector` command in Mathematica:

```
PSLQ[num_?NumericQ, basis_?VectorQ] :=
Module[{coefficients, result},
coefficients = FindIntegerNullVector[Prepend[N[basis, Precision[num]], num]];
result = Rest[coefficients].basis/First[coefficients];
Sign[N[result]] Sign[num] result;
```

The syntax is self-explanatory from the following example:

```
In[1] :=
Num = -4.2306193701686518817682268282580510275171911584045617944546633\
2782312410899782814554047544567313363330613025333597361955278613729766\
333`99.43031384975686;

Basis = {1, EulerGamma, EulerGamma^2, EulerGamma^3, EulerGamma^4, Pi^2,
Pi^4, EulerGamma Pi^2, EulerGamma^2 Pi^2, EulerGamma^3 Pi^2,
```

```

EulerGamma Pi^4, Zeta[3], Zeta[3] EulerGamma};

PSLQ[Num, Basis]

Out[1]= 1/16 (-76 - 44 EulerGamma - 22 EulerGamma^2 - \[Pi]^2
+ 14 EulerGamma \[Pi]^2 - 24 Zeta[3])

```

Above, an overcomplete basis is used for demonstration purposes only of the uniqueness of the reconstruction. A more efficient choice would require lower precision of numerical inputs.

Appendix D. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.nuclphysb.2022.116067>.

References

- [1] V.A. Smirnov, Analytic tools for Feynman integrals, Springer Tracts Mod. Phys. 250 (2012) 1–296.
- [2] S. Weinzierl, Feynman Integrals, Springer, Cham, 2022, pp. 1–857, arXiv:2201.03593 [hep-th].
- [3] C. Bogner, S. Weinzierl, Feynman graph polynomials, Int. J. Mod. Phys. A 25 (2010) 2585–2618, arXiv:1002.3458 [hep-ph].
- [4] N.I. Usyukina, A.I. Davydychev, An approach to the evaluation of three and four point ladder diagrams, Phys. Lett. B 298 (1993) 363–370.
- [5] N.I. Usyukina, A.I. Davydychev, Exact results for three and four point ladder diagrams with an arbitrary number of rungs, Phys. Lett. B 305 (1993) 136–143.
- [6] V.A. Smirnov, Analytical result for dimensionally regularized massless on shell double box, Phys. Lett. B 460 (1999) 397–404, arXiv:hep-ph/9905323.
- [7] J.B. Tausk, Nonplanar massless two loop Feynman diagrams with four on-shell legs, Phys. Lett. B 469 (1999) 225–234, arXiv:hep-ph/9909506.
- [8] A.V. Smirnov, V.A. Smirnov, On the resolution of singularities of multiple Mellin-Barnes integrals, Eur. Phys. J. C 62 (2009) 445–449, arXiv:0901.0386 [hep-ph].
- [9] M. Czakon, Automatized analytic continuation of Mellin-Barnes integrals, Comput. Phys. Commun. 175 (2006) 559–571, arXiv:hep-ph/0511200.
- [10] J.M. Henn, Multiloop integrals in dimensional regularization made simple, Phys. Rev. Lett. 110 (2013) 251601, arXiv:1304.1806 [hep-ph].
- [11] A.V. Kotikov, Differential equations method: new technique for massive Feynman diagrams calculation, Phys. Lett. B 254 (1991) 158–164.
- [12] T. Gehrmann, E. Remiddi, Differential equations for two loop four point functions, Nucl. Phys. B 580 (2000) 485–518, arXiv:hep-ph/9912329 [hep-ph].
- [13] Z. Bern, L.J. Dixon, V.A. Smirnov, Iteration of planar amplitudes in maximally supersymmetric Yang-Mills theory at three loops and beyond, Phys. Rev. D 72 (2005) 085001, arXiv:hep-th/0505205.
- [14] V.A. Smirnov, Analytical result for dimensionally regularized massless on shell planar triple box, Phys. Lett. B 567 (2003) 193–199, arXiv:hep-ph/0305142.
- [15] J.M. Henn, A.V. Smirnov, V.A. Smirnov, Analytic results for planar three-loop four-point integrals from a Knizhnik-Zamolodchikov equation, J. High Energy Phys. 07 (2013) 128, arXiv:1306.2799 [hep-th].
- [16] J.M. Henn, B. Mistlberger, Four-gluon scattering at three loops, infrared structure, and the Regge limit, Phys. Rev. Lett. 117 (2016) 171601, arXiv:1608.00850 [hep-th].
- [17] J.M. Henn, B. Mistlberger, V.A. Smirnov, P. Wasser, Constructing d-log integrands and computing master integrals for three-loop four-particle scattering, J. High Energy Phys. 04 (2020) 167, arXiv:2002.09492 [hep-ph].
- [18] J.M. Henn, K. Melnikov, V.A. Smirnov, Two-loop planar master integrals for the production of off-shell vector bosons in hadron collisions, J. High Energy Phys. 05 (2014) 090, arXiv:1402.7078 [hep-ph].
- [19] F. Caola, J.M. Henn, K. Melnikov, V.A. Smirnov, Non-planar master integrals for the production of two off-shell vector bosons in collisions of massless partons, J. High Energy Phys. 09 (2014) 043, arXiv:1404.5590 [hep-ph].

- [20] M. Beneke, V.A. Smirnov, Asymptotic expansion of Feynman integrals near threshold, Nucl. Phys. B 522 (1998) 321–344, arXiv:hep-ph/9711391.
- [21] V.A. Smirnov, Problems of the strategy of regions, Phys. Lett. B 465 (1999) 226–234, arXiv:hep-ph/9907471.
- [22] V.A. Smirnov, Applied asymptotic expansions in momenta and masses, Springer Tracts Mod. Phys. 177 (2002) 1–262.
- [23] A. Pak, A.V. Smirnov, Geometric approach to asymptotic expansion of Feynman integrals, Eur. Phys. J. C 71 (2011) 1626, arXiv:1011.4863 [hep-ph].
- [24] B. Jantzen, A.V. Smirnov, V.A. Smirnov, Expansion by regions: revealing potential and Glauber regions automatically, Eur. Phys. J. C 72 (2012) 2139, arXiv:1206.0546 [hep-ph].
- [25] A.V. Smirnov, N.D. Shapurov, L.I. Vysotsky, FIESTA5: numerical high-performance Feynman integral evaluation, Comput. Phys. Commun. 277 (2022) 108386, arXiv:2110.11660 [hep-ph].
- [26] A.V. Belitsky, V.A. Smirnov, An off-shell Wilson loop, arXiv:2110.13206 [hep-ph].
- [27] A.V. Belitsky, L.V. Bork, A.F. Pikelner, V.A. Smirnov, Exact off-shell Sudakov form factor in $N = 4$ SYM, arXiv:2209.09263 [hep-ph].
- [28] C. Anastasiou, J.B. Tausk, M.E. Tejeda-Yeomans, The on-shell massless planar double box diagram with an irreducible numerator, Nucl. Phys. B, Proc. Suppl. 89 (2000) 262–267, arXiv:hep-ph/0005328.
- [29] J. Gluza, K. Kajda, T. Riemann, AMBRE: a mathematica package for the construction of Mellin-Barnes representations for Feynman integrals, Comput. Phys. Commun. 177 (2007) 879–893, arXiv:0704.2423 [hep-ph].
- [30] M. Prausa, Mellin-Barnes meets method of brackets: a novel approach to Mellin-Barnes representations of Feynman integrals, Eur. Phys. J. C 77 (2017) 594, arXiv:1706.09852 [hep-ph].
- [31] H. Cheng, T.T. Wu, Expanding Protons: Scattering at High Energies, MIT Press, Cambridge, Massachusetts, 1987.
- [32] D.A. Kosower, BarnesRoutines.m, v.1.1.1, 2009.
- [33] G. Mishima, High-energy expansion of two-loop massive four-point diagrams, J. High Energy Phys. 02 (2019) 080, arXiv:1812.04373 [hep-ph].
- [34] H.R.P. Ferguson, D.H. Bailey, S. Arno, Analysis of PSLQ, an integer relation finding algorithm, Math. Comput. 68 (1999) 351–369.
- [35] T. Hahn, Cuba – a library for multidimensional numerical integration, Comput. Phys. Commun. 168 (2005) 78–95, arXiv:hep-ph/0404043.
- [36] M. Ochman, T. Riemann, MBsums – a Mathematica package for the representation of Mellin-Barnes integrals by multiple sums, Acta Phys. Pol. B 46 (2015) 2117, arXiv:1511.01323 [hep-ph].
- [37] A.B. Goncharov, A simple construction of Grassmannian polylogarithms, arXiv:0908.2238 [math.AG].
- [38] B. Ananthanarayan, S. Banik, S. Friot, S. Ghosh, Multiple series representations of N -fold Mellin-Barnes integrals, Phys. Rev. Lett. 127 (2021) 151601, arXiv:2012.15108 [hep-ph].
- [39] B. Ananthanarayan, S. Banik, S. Friot, T. Pathak, On the method of brackets, arXiv:2112.09679 [hep-ph].
- [40] A. Grozin, Lectures on QED and QCD: Practical Calculation and Renormalization of One- and Multi-loop Feynman Diagrams, World Scientific, 2007.