MBsums — A Mathematica PACKAGE FOR THE REPRESENTATION OF MELLIN–BARNES INTEGRALS BY MULTIPLE SUMS*

Michał Ochman

Eichwalde, Germany

Tord Riemann

Königs Wusterhausen, Germany

(Received October 19, 2015)

Feynman integrals may be represented by the Mathematica packages AMBRE and MB as multiple Mellin–Barnes integrals. With the Mathematica package MBsums we transform these Mellin–Barnes integrals into multiple sums.

DOI:10.5506/APhysPolB.46.2117
PACS numbers: 11.80.Cr, 12.38.Bx

1. Introduction

In recent years, there was a remarkable progress in the development of (semi-)automatized software for the numerical calculation of arbitrary, complicated Feynman integrals. Basically, two approaches are advocated. One relies on a sector decomposition. For an introduction given at this conference and for further references see [1]. We will report on the other approach, based on Mellin–Barnes representations [2–12]. When [9] appeared in 2005, several unsolved problems of different complexity existed. We mention non-planar diagrams, the massive case, multi-loop tensor integrals, Minkowskian kinematics. For all of the items, a progress is reported in [2], based on the source-open software AMBRE/MB [8, 13, 14]. An alternative is the direct analytical evaluation of MB-integrals. This is difficult. But in view of the recent progress in algebraically summing up infinite sums by the Linz group's computer algebra algorithms for nested sums and products, one might hope

* Presented by J. Gluza at the XXXIX International Conference of Theoretical Physics "Matter to the Deepest", Ustroń, Poland, September 13–18, 2015.

to achieve a breakthrough [15]; certainly only if the result leads to appropriate classes of functions. The idea is to apply the Linz group's algorithms (as well as those of others, *e.g.* [16]) to sums of residues after applying Cauchy's theorem [17]. A first attempt was reported in [18].

The automatized derivation of the multiple sums for a given MB-integral is certainly the easier part of the task, but it is the first step. We report here on a first version of the Mathematica program MBsums [19] for transforming MB-representations for Feynman integrals into multiple sums. The licence conditions of the source-open package are those formulated in the CPC non-profit use licence agreement of the Computer Physics Communications Program Library [20]. The authors expect that the potential users read and follow the licence agreement when using this code.

2. The Mathematica package MBsums

The package MBsums transforms Mellin–Barnes integrals into sums, by closing the integration contours and calculating the integrals by the residue theorem, *i.e.* by constructing sums over all residues inside the contours. The current version of MBsums is 1.0. The package MBsums works with Wolfram Mathematica 7.0 and later.

In order to obtain a sum from an MB-integral, the user should use the MBIntToSum function of MBsums:

`MBIntToSum[int,{},contours]` (1)

or

`MBIntToSum[int,kinematics,contours]` , (2)

where `int` is the MB-integral in the form as it is denoted in the Mathematica package `MB` [9][1]:

`int` = `MBInt[f,{{eps -> 0},{z1 -> c1, z2 -> c2, ... , zD -> cD}]` (3)

which corresponds to

$$\texttt{int} = \frac{1}{(2\pi i)^d} \int\limits_{-i\infty+c_1}^{i\infty+c_1} \cdots \int\limits_{-i\infty+c_D}^{i\infty+c_D} f \prod_{k=1}^{D} \mathrm{d}z_k \,. \tag{4}$$

The integrand f can have the form

$$f = \sum_j f_j \tag{5}$$

[1] The package MBsums uses notations of the package MB, but can be run also independently.

and each f_j is assumed to be of the form of

$$f_j = \xi_j \frac{\prod_m \Gamma(N_m^{(j,1)}) \prod_m \Psi^{(n)}\left(N_m^{(j,2)}\right)}{\prod_m \Gamma\left(N_m^{(j,3)}\right)} \prod_m r_m^{N_m^{(j,4)}}, \tag{6}$$

where $\Psi^{(0)}(z) = d\ln(\Gamma(z))/dz$, $\Psi^{(n>0)}(z) = d^n\Psi^{(0)}(z)/dz^n$, r_i are free parameters (usually kinematic parameters) in `int` and ξ_j is a factor independent of z-variables. The

$$N_m^{(j,k)} = \sum_i \alpha_{m,i}^{(j,k)} z_i + \beta_m^{(j,k)} + \gamma_m^{(j,k)}\epsilon\,, \tag{7}$$

where $\alpha_{m,i}^{(j,k)}$, $\beta_m^{(j,k)}$, $\gamma_m^{(j,k)}$ are rational numbers and ϵ (`eps`) is an infinitesimal dimensional shift, *e.g.* arising from $d = 4 - 2\epsilon$. All the singularities of the integrand of the MB-integral `f` are due (and only due) to `Gamma` and `PolyGamma` functions.

The values of `c1, c2, ... , cD` are converted to rational numbers by `MBIntToSum` before calculations.

Let us now focus on the case when the list `kinematics` is empty, *i.e.* we will consider (2). The list `contours` has the form

`contours` = `{z1 -> L/R, z2 -> L/R, ... , zD -> L/R}`. (8)

The order of the z-variables defines the order of integrations chosen by the user (from left to right). The `L` (`R`) means that the contour will be closed to the left (right). The `L`/`R` choice made by the user can be changed if `kinematics` is not empty and this will be covered later. The output of `MBIntToSum` in (2) is of the form

`{MBsum_1, MBsum_2, ... , MBsum_Q}`, (9)

where

`MBsum_i` = `MBsum[Sum_Coefficient_i,Conditions_i,List_i]` (10)

represents a sum with summand `Sum_Coefficient_i` that has non-negative indices given in the list `List_i`, and `Conditions_i` are conditions on those indices. The complete answer is the sum of all `MBsum_i` in the list.

The list `kinematics` has the form

`kinematics` = `{r_1 -> v_1, r_2 -> v_2, ... , r_K -> v_K}`, (11)

where `r_i` are free parameters (usually kinematic parameters) in `int` and `v_i` are values of `r_i`. If `kinematics` is not empty, then `MBIntToSum` will try to change the `L`/`R` choice made by the user in the list `contours` in order to obtain sums that have good asymptotic behaviour at given values of `r_i`. In any case the user is informed how the contour was closed. This will be explained later in detail. The values of `v_i` are converted to rational numbers by `MBIntToSum` before calculations. The user can turn off all messages printed by `MBIntToSum` by typing `MBsumsInfo=False` and turn them on by typing `MBsumsInfo=True`.

In addition, we provide function `DoAllMBSums[sums,nmax,kinematics]` that sums the `sums` in the form of (9). The `nmax` is the maximal value of each index, the minimal value is given by conditions on indices. The list `kinematics` is as above and may be empty. We used Wolfram Mathematica function `ParallelMap` inside `DoAllMBSums` to sum individual sums in the list `sums` in parallel.

3. Obtaining the sums

Let us now shortly explain how we obtain the sums. We point out the most important ingredients in our algorithm. Let us focus on the case when the list `kinematics` is empty, *i.e.* we will consider (2). The MB-integral is in the form as it was denoted in (3). Let us now assume that the user has chosen as first integration the `z2->L`. As a first step we form a list, which we call NegArgsDoC, of arguments of the `Gamma` and `PolyGamma` functions in the numerators that give residues for $\mathrm{Re}(z2) < c2$ (see (3), the remaining contours are seen as a straight lines). We call that list NegArgsDoC. Next, we consider all possible cases: When all `Gamma` and `PolyGamma` functions that have arguments in NegArgsDoC contribute to a residue at the same time, and when only some subset of them contributes to a residue at the same time. We consider all possible subsets of NegArgsDoC. Additionally, we have to be careful when some `Gamma` functions in the denominator become singular at some points. If we have terms like `Gamma[2 z2]`, then the poles are at $z2 = -n/2$ and we consider there 2 cases: $n = 2n'$ and $n = 2n' + 1$, where n and n' are non-negative integers. Similarly, we proceed with arbitrary $M \times z2$ terms, where M is some integer value and, in general, with all n/M terms which appear together with integration variable in the arguments of the `Gamma` and `PolyGamma` functions. So we produce a list of cases

$$\left\{\left\{f_1^{(1)}, c_1^{(1)}\right\}, \left\{f_2^{(1)}, c_2^{(1)}\right\}, ..., \left\{f_{K_1}^{(1)}, c_{K_1}^{(1)}\right\}\right\}, \tag{12}$$

where $f_i^{(1)}$ are expressions after taking residues of f and $c_i^{(1)}$ are conditions on the index that numerates terms (residues) in $f_i^{(1)}$. We obtain a list of K_1 elements after integrating over $z2$. Let us now assume that the user has chosen as second integration variable `z5->R`. Then, we repeat the whole procedure on each $f_i^{(1)}$ taking into account conditions $c_i^{(1)}$. Thus, we produce analogous to (12) a list of cases

$$\left\{\left\{f_1^{(2)},c_1^{(2)}\right\},\left\{f_2^{(2)},c_2^{(2)}\right\},...,\left\{f_{K_2}^{(2)},c_{K_2}^{(2)}\right\}\right\}. \tag{13}$$

We repeat the whole procedure for each integration variable.

4. Contours and convergent sums

Let us now shortly explain how we obtain the sums if the list `kinematics` is not empty. We follow the order of integration given in the list `contours`. Our aim is to determine the `L`/`R` such that we obtain sums that have good asymptotic behaviour at given values of `r_i` in the list `kinematics`. We do it in the following way. At each integration step s, we analyse the expressions $f_i^{(s)}$ that are to be integrated over some z_C. Each $f = f_i^{(s)}$ we decompose as

$$f = \sum_j g_j \tag{14}$$

and each $g = g_j$ is of the form of

$$g = r_1^{a_1} r_2^{a_2} \ldots r_K^{a_K} F \,, \qquad a_j = \sum_i a_{j,i} z_i \,, \tag{15}$$

where r_i are the kinematic parameters in the list `kinematics` and F contains the rest of g. If we integrate over z_C, we consider

$$c^{z_C} \,, \qquad c = r_1^{a_{1,C}} r_2^{a_{2,C}} \ldots r_K^{a_{K,C}} \,. \tag{16}$$

The value of c is calculated. `MBIntToSum` prints the error message:

`Found c =` c `(not a number): please complete kinematic's list`

for each g_i in each $f_i^{(s)}$ when c is symbolic (not a number) and at the end `MBIntToSum` prints

`Unable to find correct contour for` z_C

and returns {}. The user should complete the list `kinematics`.

For each g_i in each $f_i^{(s)}$ it is returned `L` if $|c| > 1$ or `R` if $|c| < 1$ indicating how to close the contour or {} if $|c| = 1$.

If for each g_i in each $f_i^{(s)}$ it is returneded `L` (`R`) or {}, then the contour for z_C will be closed to left (right) if it is returned at least one `L` (`R`).

If for each g_i in each $f = f_i^{(s)}$ it is returned {}, then the choice of user given in the list `contours` is taken.

If for some g_i it is returned `R` and for some g_j it is returned `L`, then `MBIntToSum` prints the error message:

`Unable to find correct contour for` z_C

and returns {}. Otherwise, we compute the sums as described above. We repeat the whole procedure for each integration variable. We stress that this procedure as described above does not always give convergent sums.

There are MB-integrals for which no convergent sums can be found. One such example is the following MB-integral:

$$B_1 = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} dz \Gamma^2(1+z)\Gamma^2(1-z)\,. \tag{17}$$

Here we can apply the first Barnes lemma [3] and obtain $B_1 = 1/6$, but the reader can check that indeed the infinite series of residues diverge both for $\mathrm{Re}\, z > 0$ and $\mathrm{Re}\, z < 0$.

Consider the following integral (see also [3]):

$$B_x = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} dz x^z \Gamma^2(1+z)\Gamma^2(1-z)\,, \qquad x > 0 \wedge x \neq 1\,. \tag{18}$$

Closing the contour to the right ($\mathrm{Re}\, z > 0$) gives the following series

$$s_{\mathrm{R}} = -\sum_{n=1}^{\infty} n x^n (2 + n\ln(x))\,, \tag{19}$$

convergent for $0 < x < 1$, while closing the contour to the left ($\mathrm{Re}\, z < 0$) gives the following series

$$s_{\mathrm{L}} = -\sum_{n=1}^{\infty} n x^{-n} (2 - n\ln(x))\,, \tag{20}$$

convergent for $x > 1$. Both s_{L} and s_{R} give the same formula after summing up, that is

$$s_{\mathrm{LR}} = \frac{x(2 - 2x + (1+x)\ln(x))}{(x-1)^3}\,, \qquad x > 0 \wedge x \neq 1\,, \tag{21}$$

so $B_x = s_{\mathrm{LR}}$ and $\lim_{x\to 1} B_x = B_1$.

We would like to thank Johannes Blümlein, Ievgen Dubovyk, Janusz Gluza and Carsten Schneider for numerous discussions. This work summarizes a part of a talk given by J. Gluza at the "Matter to the Deepest 2015" conference. The write-up has been shortened for the text version to appear in *Acta Physica Polonica B*. Work is supported by the Polish National Science Centre (NCN) under the Grant Agreement No. DEC-2013/11/B/ST2/0402.

REFERENCES

[1] S. Borowka, *Acta Phys. Pol. B* **46**, 2137 (2015), this issue.

[2] J. Gluza, "Mellin–Barnes Representations of Feynman Integrals: a Construction and Solutions", talk given at this conference `http://indico.if.us.edu.pl/getFile.py/access?contribId=121&sessionId=&resId=0&materialId=slides&confId=2`

[3] E.W. Barnes, *Proc. Lond. Math. Soc. (2)* **6**, 141 (1908).

[4] N. Usyukina, *Teor. Mat. Fiz.* **22**, 300 (1975).

[5] E. Boos, A.I. Davydychev, *Theor. Math. Phys.* **89**, 1052 (1991).

[6] V.A. Smirnov, *Phys. Lett. B* **460**, 397 (1999) [arXiv:hep-ph/9905323].

[7] J. Tausk, *Phys. Lett. B* **469**, 225 (1999) [arXiv:hep-ph/9909506].

[8] MB Tools webpage, `http://projects.hepforge.org/mbtools/`

[9] M. Czakon, *Comput. Phys. Commun.* **175**, 559 (2006) [arXiv:hep-ph/0511200].

[10] A.V. Smirnov, V.A. Smirnov, *Eur. Phys. J. C* **62**, 445 (2009) [arXiv:0901.0386 [hep-ph]].

[11] J. Gluza, K. Kajda, T. Riemann, *Comput. Phys. Commun.* **177**, 879 (2007) [arXiv:0704.2423 [hep-ph]].

[12] J. Gluza, K. Kajda, T. Riemann, V. Yundin, *Eur. Phys. J. C* **71**, 1516 (2011) [arXiv:1010.1667 [hep-ph]].

[13] I. Dubovyk, J. Gluza, T. Riemann, *J. Phys. Conf. Ser.* **608**, 012070 (2015).

[14] Silesian University, Katowice, webpage `http://prac.us.edu.pl/~gluza/ambre`

[15] C. Schneider *et al.* [RISC Combinatorics Group], Research Institute for Symbolic Computation, J. Kepler University, Linz, `https://www.risc.jku.at/people/cschneid/`

[16] S. Moch, P. Uwer, *Comput. Phys. Commun.* **174**, 759 (2006) [arXiv:math-ph/0508008].

[17] J. Gluza, T. Riemann, "Simple Feynman Diagrams and Simple Sums", talk held at the RISC-DESY Workshop on Advanced Summation Techniques and their Applications in Quantum Field Theory, on the occasion of the 5$^{\text{th}}$ year jubilee of the RISC-DESY cooperation, May 7–8, 2012, RISC Institute, Castle of Hagenberg, Linz, Austria. Conference link: `http://www.risc.jku.at/conferences/RISCDESY12/`, talk link: `http://www-zeuthen.desy.de/~riemann/Talks/riemann-Linz-2012-05.pdf`

[18] J. Blümlein *et al.*, *PoS* **LL2014**, 052 (2014) [arXiv:1407.7832 [hep-ph]].

[19] M. Ochman, Mathematica package MBsums v.1.0, available at [14].

[20] The CPC non-profit use licence agreement of the Computer Physics Communications Program Library (7 Oct 2015), `http://cpc.cs.qub.ac.uk/licence/licence.html`