

# Solving differential equations for Feynman integrals by expansions near singular points

Roman N. Lee,<sup>a</sup> Alexander V. Smirnov<sup>b</sup> and Vladimir A. Smirnov<sup>c</sup>

<sup>a</sup>*Budker Institute of Nuclear Physics,  
630090 Novosibirsk, Russia*

<sup>b</sup>*Research Computing Center, Moscow State University,  
119991, Moscow, Russia*

<sup>c</sup>*Skobeltsyn Institute of Nuclear Physics of Moscow State University,  
119991, Moscow, Russia*

*E-mail:* [roman.n.lee@gmail.com](mailto:roman.n.lee@gmail.com), [asmirnov80@gmail.com](mailto:asmirnov80@gmail.com),  
[smirnov@theory.sinp.msu.ru](mailto:smirnov@theory.sinp.msu.ru)

**ABSTRACT:** We describe a strategy to solve differential equations for Feynman integrals by powers series expansions near singular points and to obtain high precision results for the corresponding master integrals. We consider Feynman integrals with two scales, i.e. non-trivially depending on one variable. The corresponding algorithm is oriented at situations where canonical form of the differential equations is impossible. We provide a computer code constructed with the help of our algorithm for a simple example of four-loop generalized sunset integrals with three equal non-zero masses and two zero masses. Our code gives values of the master integrals at any given point on the real axis with a required accuracy and a given order of expansion in the regularization parameter  $\epsilon$ .

**KEYWORDS:** Scattering Amplitudes, Perturbative QCD

**ARXIV EPRINT:** [1709.07525](https://arxiv.org/abs/1709.07525)

---

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                                       | <b>1</b>  |
| <b>2</b> | <b>Generalized series expansion near a singular point</b> | <b>3</b>  |
| <b>3</b> | <b>Matching</b>   | <b>6</b>  |
| <b>4</b> | <b>Implementation</b>                                     | <b>7</b>  |
| <b>5</b> | <b>Conclusion</b>   | <b>11</b> |

---

## 1 Introduction

Evaluating Feynman integrals with differential equations (DE) initiated in [1, 2] and formulated as a method to evaluate master integrals in [3–6] became one of the most powerful methods already much time ago. Still this method is under development. In [7] it was suggested to turn from the basis of primary master integrals (i.e. revealed when solving integration by parts relations [8]) to the so-called *canonical* basis for which the right-hand side of the of system of DE is just proportional to  $\epsilon = (4 - D)/2$  and the singularities of the matrix on the right-hand side of DE are Fuchsian, i.e. have only simple poles in all the singular points of DE. The first algorithm to arrive at the canonical form was constructed in the case of one variable in ref. [9] (Such form of DE was called  $\epsilon$ -form there). Besides the private implementation of this algorithm by its author and several other private implementations, two public implementations, `Fuchsia` [10, 11] and `epsilon` [12], of the algorithm of ref. [9] are now available.<sup>1</sup>

Once DE for master integrals are converted into an  $\epsilon$ -form, i.e. one finds an appropriate linear transformation to a canonical basis, solving DE becomes straightforward, order-by-order in  $\epsilon$ , at least in the form of iterated integrals, with possible complications connected with letters expressed in terms of square roots. Typically, the corresponding results are expressed naturally in terms of harmonic polylogarithms [15] or multiple polylogarithms [16]. These functions are very well studied. For harmonic polylogarithms, one can apply the package `HPL` [17] which encodes various analytical properties and provides the possibility of numerical evaluation with a very high precision. For multiple polylogarithms, one can use the computer code [18] based on the `GiNaC` library [19] to obtain high-precision numerical values, up to several thousand digits and more.

It is well known that the  $\epsilon$ -form of DE for a given set of the master integrals is not always achievable by rational transformations. For massive internal lines it is often required to consider also transformations involving square roots. However, even using

---

<sup>1</sup>See also [13, 14] where an algorithm in the case of two and more variables is described and implemented.

transformations from this extended class it is not always possible to obtain an  $\epsilon$ -form.<sup>2</sup> The simplest example where an  $\epsilon$ -form is impossible is given by the two-loop propagator sunset diagram with three identical masses. In this example, as well in other known examples without  $\epsilon$ -form, DE can still be reduced to the form where the right-hand side of the differential system is a linear function of  $\epsilon$ .

However, ‘integrating out’ the constant term in such a form of DE appears to be an essentially more complicated problem. This can be seen in the known examples where results are expressed in terms of elliptic functions. In practice, it can happen that such ‘elliptic’ master integrals appear only in a small number of sectors. (A sector is specified by a distribution of the set of indices (powers of propagators) into positive and non-positive values.) A first example of a calculation of a full set of the master integrals with ‘elliptic sectors’ can be found in ref. [21]. A recent example of a calculation relevant to a physical problem is ref. [22], where elliptic functions appear only in two sectors and final results are expressed either in terms of multiple polylogarithms or, for the elliptic sectors and for some sectors involving complicated letters with square roots, in terms of two and three-fold iterated integrals suitable for numerical evaluation. Moreover, in refs. [23, 24] a strategy to obtain parametric representations for master integrals applicable also in situations without  $\epsilon$ -form was described and illustrated through one-, two- and three-loop examples.

Other examples of calculations of individual Feynman integrals in situations where  $\epsilon$ -form is impossible can be found in [25, 26] (see also references therein), where results are expressed in terms of elliptic generalizations of polylogarithms [25] or iterated integrals of modular forms [26]. One more class of elliptic generalization of multiple polylogarithms was recently introduced in ref. [27]. In particular, it includes functions appearing in the  $\epsilon$ -expansion of the imaginary part of the two-loop massive sunset diagram. However, these new functions do not have the same status as harmonic polylogarithms and multiple polylogarithms, at least in the practical sense, i.e. there are no codes to evaluate them at a given point with a desired precision. Anyway, it looks like we are very far, even in lower loops orders, from answering the following question: ‘What is the class of functions which can appear in results for Feynman integrals in situations where  $\epsilon$ -form is impossible’?

On the other hand, thinking positively, we may say that knowing a differential system and the corresponding boundary conditions gives almost as much information about Feynman integrals as knowing their explicit expressions in terms of some class of functions. In fact, some properties of the integrals are even more accessible via DE. In particular, singularities of DE provide a way to examine the branching properties of integrals. Numerical values of the integrals can be obtained from a numerical solution<sup>3</sup> of the differential system. Many computer algebra systems contain tools to solve this task (e.g. `NDSolve` procedure in *Mathematica* system). However, there is one complication that does not allow to use these tools immediately. Namely, we would like to keep  $\epsilon$  as a variable and evaluate solutions of DE as series expansions in  $\epsilon$ .

---

<sup>2</sup>Recently, a strict criterion of the existence of an  $\epsilon$ -form was presented in ref. [20].

<sup>3</sup>Examples of solving DE for Feynman integrals numerically can be found in refs. [28, 29].

The goal of the present paper is to describe an algorithm which enables one to find a solution of a given differential system in the form of an  $\epsilon$ -expansion series with numerical coefficients. We describe such an algorithm in the case of Feynman integrals depending on one variable, i.e. with two scales where the variable is introduced as the ratio of these scales. As a proof of concept, we provide a computer code where this algorithm is implemented for a simple example of a family of Feynman integrals where the  $\epsilon$ -form is impossible. The general idea behind our approach is to use generalized power series expansions near the singular points of the differential system and solve difference equations for the corresponding coefficients in these expansions. This idea is very well known in mathematics: already in the classical textbook [34] the algorithm to obtain first terms of expansion near a singular point is described.<sup>4</sup> In high-energy physics, its application to Feynman integrals can be found, for example, in refs. [21, 30–32, 35] to evaluate expansions of solutions of DE at a given singular point by difference equations.

Though the approach of the present paper follows approximately the same lines, there are some peculiarities connected with our goals. Namely, we are aimed at the automatic calculation of multiloop integrals at an arbitrary kinematic point up to an arbitrary order in  $\epsilon$  with an arbitrary precision. It is the latter goal that requires effective automatic generation of many terms of power series expansion ‘on the fly’. In the next section, we present an algorithm to solve difference equations for coefficients of the series expansions at a given singular point. The advantage of the presented algorithm is that its computational complexity grows only linearly with the required number of terms. In section 3, we describe a matching procedure which enables one to connect series expansions at two neighboring points. In section 4, we describe a `Mathematica` code based on our algorithm and the matching procedure to evaluate master integrals in a simple four-loop example. This code as well as the corresponding auxiliary files can be downloaded from <https://bitbucket.org/feynmanintegrals/dess>. Then we conclude with a discussion of perspectives.

## 2 Generalized series expansion near a singular point

Let us have a differential system

$$\partial_x \mathbf{J} = M(x, \epsilon) \mathbf{J}, \quad (2.1)$$

where  $\mathbf{J}$  is a column-vector of  $N$  functions, and  $M$  is an  $N \times N$  matrix with entries being rational functions<sup>5</sup> of  $x$  and  $\epsilon$ . Below we will suppress  $\epsilon$  in the arguments for brevity. We assume that all the singular points of the differential system are regular. This implies that we can reduce the differential system to a local Fuchsian form in any singular point.

The general solution of this linear system has the form

$$\mathbf{J}(x) = U(x) \mathbf{J}_0, \quad (2.2)$$

---

<sup>4</sup>This algorithm was used, in particular, in ref. [33].

<sup>5</sup>Typically,  $x$  is a dimensionless ratio of two scales for a family of dimensionally regularized Feynman integrals depending on two scales.

where  $\mathbf{J}_0$  is a column of constants, and  $U$  is an evolution operator represented in terms of a path-ordered exponential

$$U(x) = P \exp \left[ \int dx M(x) \right]. \tag{2.3}$$

We want to expand this operator in the vicinity of each singular point. Without loss of generality, let us consider the expansion near  $x = 0$ . It is well known that the expansion has the form

$$U(x) = \sum_{\lambda \in S} x^\lambda \sum_{n=0}^{\infty} \sum_{k=0}^{K_\lambda} \frac{1}{k!} C(n + \lambda, k) x^n \ln^k x, \tag{2.4}$$

where  $S$  is a finite set of powers,  $K_\lambda \geq 0$  is an integer number corresponding to the maximal power of the logarithm. We have introduced the factor  $1/k!$  for convenience. Our goal is to determine  $S$ ,  $K_\lambda$ , and the matrix coefficients  $C(n + \lambda, k)$ . As to the latter, we are going to determine them via recurrence relations equipped with initial conditions.

Since we assume that the differential system has only regular singular points, we can reduce it at  $x = 0$  to normalized Fuchsian form [20] by means of rational transformations. For the sake of presentation, we will assume that the system is in global normalized Fuchsian form, i.e.,

$$M(x) = \frac{A_0}{x} + \sum_{k=1}^s \frac{A_k}{x - x_k} \tag{2.5}$$

and for any  $k = 0, \dots, s$  the matrix  $A_k$  is free of resonances, i.e. the difference of any two of its distinct eigenvalues is not integer. Note that the  $\epsilon$ -form is only one example of normalized Fuchsian form, so we allow for a much wider class of differential systems which seems to be sufficient for any applications in multiloop calculations. In particular, the ‘elliptic’ cases, as a rule, can easily be reduced to a global normalized Fuchsian form. Besides, it is easy to generalize our algorithm properly if needed.

In order to obtain a recurrence relation of a finite order, we will first multiply both sides of eq. (2.1) by the common denominator  $xQ(x)$ , where

$$Q(x) = \prod_{k=1}^s (x - x_k) = \sum_{m=0}^s q_m x^m. \tag{2.6}$$

By construction we have  $q_0 \neq 0$ . We will also define the polynomial matrix  $B(x, \alpha)$  and its coefficients  $B_m(\alpha)$  by

$$B(x, \alpha) = Q(x) (xM(x) - \alpha) = \sum_{m=0}^s B_m(\alpha) x^m. \tag{2.7}$$

Note that  $B_0(\alpha) = q_0(A_0 - \alpha)$ .

Then the recurrence relations read

$$\begin{aligned} & - \text{BJF}(B_0(\lambda + n), -q_0, K_\lambda) C(\lambda + n, 0 \dots K_\lambda) \\ & = \sum_{m=1}^s \text{BJF}(B_m(\lambda + n - m), -q_m, K_\lambda) C(\lambda + n - m, 0 \dots K_\lambda). \end{aligned} \tag{2.8}$$

Here  $C(\alpha, 0 \dots K) = \begin{bmatrix} C(\alpha, 0) \\ \vdots \\ C(\alpha, K) \end{bmatrix}$  denotes a  $(K + 1)N \times N$  matrix built from blocks  $C(\alpha, k)$  and the three-letter notation BJF stands for ‘Block Jordan Form’ defined as

$$\text{BJF}(A, B, K) = \underbrace{\begin{bmatrix} A & B & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & B \\ 0 & 0 & 0 & A \end{bmatrix}}_{K+1}.$$

Now note that the operator  $U$ , eq. (2.3), is determined up to a multiplication by a constant matrix from the right. We fix it by the condition

$$U(x) \stackrel{x \rightarrow 0}{\sim} x^{A_0}. \tag{2.9}$$

This condition is, strictly speaking, mathematically incorrect when the distance between some eigenvalues of  $A_0$  is larger than one, but it should be understood as the constraint on the leading terms of the expansion for each distinct eigenvalue. This condition gives us a way to determine  $S$ , i.e. the set of distinct eigenvalues of  $A_0$ , and  $K_\lambda$ , i.e. the highest power of the logarithm in front of  $x^\lambda$  in  $x^{A_0}$  for each  $\lambda \in S$ , and the leading coefficients  $C(\lambda, k)$ . We simply determine these parameters by representing

$$x^{A_0} = \sum_{\lambda \in S} x^\lambda \sum_{k=0}^{K_\lambda} \frac{1}{k!} C(\lambda, k) \ln^k x. \tag{2.10}$$

Now note that the matrix  $-\text{BJF}(B_0(\lambda + n), -q_0, K_\lambda)$  on the left-hand side of eq. (2.8) is invertible for  $\lambda \in S$  and  $n > 0$ . Indeed,

$$\det \text{BJF}(B_0(\lambda + n), -q_0, K_\lambda) = (\det B_0(\lambda + n))^{K_\lambda+1} = q_0^{(K_\lambda+1)n} [\det(A_0 - \lambda - n)]^{K_\lambda+1} \tag{2.11}$$

and both  $q_0 \neq 0$  and  $\det(A_0 - \lambda - n) \neq 0$ , the latter is due to the absence of resonances in  $A_0$  (since if  $\det = 0$  both  $\lambda$  and  $\lambda + n$  would be the eigenvalues of  $A_0$ ). Therefore, we can rewrite recurrence relations (2.8) as

$$C(\lambda + n, 0 \dots K_\lambda) = \sum_{m=1}^s T(\lambda, n, m) C(\lambda + n - m, 0 \dots K_\lambda), \tag{2.12}$$

where

$$T(\lambda, n, m) = -[\text{BJF}(B_0(\lambda + n), -q_0, K_\lambda)]^{-1} \text{BJF}(B_m(\lambda + n - m), -q_m, K_\lambda). \tag{2.13}$$

and use (2.12) together with the initial conditions determined<sup>6</sup> by eq. (2.10) in order to construct the generalized power series expansion (2.4). Note that the finite-order recurrence relation results in a linear growth of the computational complexity with the number of expansion terms.

---

<sup>6</sup>One also puts  $C(\lambda + n, k) = 0$  for  $\lambda \in S$  and  $n < 0$ .

To summarize, the data necessary to obtain the expansion (2.4) are as follows:

1. The set  $S = \{\lambda_1, \lambda_2, \dots\}$  of the eigenvalues of the matrix residue  $A_0$ .
2. For each  $\lambda \in S$ :
  - (a) the maximal power of the logarithm  $K_\lambda$  and the leading coefficients  $C(\lambda, 0 \dots K_\lambda)$  defined by (2.10). To use the recurrence formula one has to take into account that  $C(\lambda + n, k) = 0$  for  $n < 0$ .
  - (b) the matrix coefficients  $T(\lambda, n, 1), \dots, T(\lambda, n, s)$  which are  $(K_\lambda + 1)N \times (K_\lambda + 1)N$  matrices, where the dependence on  $n$  is explicit.

**Expansion in  $\epsilon$ .** So far we paid no attention to  $\epsilon$  dependence of both the eigenvalues  $\lambda_i$  and the coefficients  $C(\lambda + n, k)$ . Fortunately, the question of how to expand obtained formulas in  $\epsilon$  is very simple. Taking into account that the coefficients  $C(\lambda + n, k)$  are always rational functions of  $\epsilon$  and the convergence of power series is determined by the factor  $x^n$ , it is easy to conclude that the  $\epsilon$ -expansion commutes with the summation. Therefore one can simply expand the coefficients  $C(\lambda + n, k)$  under the sum. As we should treat  $\epsilon$  as the ‘smallest’ parameter, we have  $|\epsilon \log(x)| \ll 1$  and therefore, we also expand the factors  $x^{\lambda_i}$ .

### 3 Matching

The above considerations enable one to evaluate the evolution operator (2.3) within the convergence region of the power series (2.4). In order to perform an analytical continuation to the whole complex plane, one may use the same approach for the expansion around other singular points. Suppose that the next singular point closest to the origin is  $x = 1$ . We can construct the evolution operator (2.3) also in an expansion near this point,

$$\tilde{U}(x) = P \exp \left[ \int dx M(x) \right]. \tag{3.1}$$

In general, due to the above mentioned freedom in the definition of the evolution operator we have

$$U(x) = \tilde{U}(x) L.$$

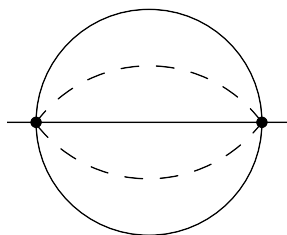
where  $L$  is some constant matrix. If the convergence regions of the power series in  $U$  and  $\tilde{U}$  overlap, we may fix  $L$  by picking some point in the intersection of these regions. E.g. at  $x = 1/2$  we have<sup>7</sup>  $L = \tilde{U}^{-1}(1/2) U(1/2)$ , i.e., finally, in the whole convergence region of  $\tilde{U}$  we have

$$U(x) = \tilde{U}(x) \tilde{U}^{-1}(1/2) U(1/2).$$

Acting in the same way, we may, in principle, extend the definition of  $U$  onto the whole complex plane of  $x$ . In fact, this is a general approach to the analytical continuation

---

<sup>7</sup>The convergence radius of the power series is equal to the distance to the closest singularity, so  $x = 1/2$  necessarily belongs to the convergence region of the series representation of  $U$ . We also assume here that it belongs to the convergence region of  $\tilde{U}$ .



**Figure 1.** The generalized sunset graph with two massless and three massive lines with the same mass.

of a function defined by a converging power series. In order to reach an arbitrary finite point of the complex plane, we are likely to need also expansions near the regular points (reducible to the considered case by putting  $A_0 = 0$ ) and/or Möbius transformations of the variable. In the case where the singularities lie on the real axis and if we are interested in the evaluation of Feynman integrals for real  $x$ , we can avoid expansions near regular points and rely only on the Möbius transformations. Suppose, e.g., that we have the following sequence of the singular points

$$x_0 < x_1 < \dots x_s < \infty = x_{s+1} = x_{-1}. \tag{3.2}$$

Then for each  $0 \leq k \leq s$  we make the variable change

$$y_k(x) = \frac{ax + b}{cx + d} \tag{3.3}$$

which maps the points  $x_{k-1}, x_k, x_{k+1}$  to  $\mp 1, 0, \pm 1$ , respectively.<sup>8</sup> It is convenient to choose the sign in such a way that the cuts of the non-integer powers and logarithms appearing in the series expansions coincide with the cuts of the integral.

## 4 Implementation

The four master integrals we evaluate form a basis of the following family of integrals:

$$F_{a_1, \dots, a_{14}} = \int \dots \int \frac{d^D k_1 \dots d^D k_4 (k_1 \cdot p)^{a_6} (k_2 \cdot p)^{a_7} (k_3 \cdot p)^{a_8} (k_4 \cdot p)^{a_9}}{(-k_1^2)^{a_1} (-k_2^2)^{a_2} (m^2 - k_3^2)^{a_3} (m^2 - k_4^2)^{a_4} (m^2 - (\sum k_i + p)^2)^{a_5}} \times (k_1 \cdot k_2)^{a_{10}} (k_1 \cdot k_3)^{a_{11}} (k_1 \cdot k_4)^{a_{12}} (k_2 \cdot k_3)^{a_{13}} (k_2 \cdot k_4)^{a_{14}}, \tag{4.1}$$

where  $p$  is the external momentum and  $m$  is the mass of three lines. They correspond to the generalized sunset graph shown in figure 1. We introduce  $x = p^2/m^2$ .

There are four master integrals in this family. As the primary master integrals we choose the following basis:

$$\mathbf{J}_1 = \{F_{1,1,1,1,1,0,\dots,0}, F_{1,1,2,1,1,0,\dots,0}, F_{1,2,1,1,1,0,\dots,0}, F_{1,2,1,1,2,0,\dots,0}\}. \tag{4.2}$$

---

<sup>8</sup>Explicitly we have  $y_k(x) = \pm \frac{(x-x_k)(x_{k+1}-x_{k-1})}{(x-x_{k+1})(x_{k-1}-x_k) + (x-x_{k-1})(x_{k+1}-x_k)}$ .



We derive DE for  $\mathbf{J}_1$  in a straightforward way. When taking derivatives with respect to  $x$  one can apply `LiteRed` [39, 40] to do this automatically. The derivatives are then expressed in terms of integrals of the given family. Solving integration by parts relations with an IBP-reduction code,<sup>9</sup> one expresses these derivatives as linear combinations of the primary master integrals and obtains a system of linear DE which has the form (2.1).

The matrix in the corresponding DE, as well as other entries connected with the described `Mathematica` code, can be downloaded from <https://bitbucket.org/feynmanintegrals/dess>. One uses

```
{M, T, Ti, Mf} = << "Data/TransformationData";
```

Here  $\mathbf{M} = M$  is the matrix on the right-hand side of the original differential system,  $\mathbf{Mf} = M_f$  is the matrix of the transformed system, and  $\mathbf{T} = T$  and  $\mathbf{Ti} = T^{-1}$  are the transformation matrix and its inverse.<sup>10</sup> We turn to the basis  $\mathbf{J} = T^{-1} \cdot \mathbf{J}_1$  for which we have the matrix  $M_f = T^{-1}(M \cdot T - \partial_x T)$  with normalized Fuchsian singularities at any singular point in the corresponding DE (2.1). The transformation  $T$  to the new basis was found with the help of the algorithm of ref. [9].

To fix boundary conditions we choose the point  $x = 0$  where the integrals of the given family become vacuum integrals. To evaluate the four master integrals at  $x = 0$  we derive onefold Mellin-Barnes representations for them and obtain the possibility to achieve a high precision for any given coefficient in the  $\epsilon$ -expansion. We restricted ourselves to the accuracy of 500 digits but one can increase it to 1000 digits and more.

The singular points are  $x_0 = 0, x_1 = 1, x_2 = 9$  and  $x_3 = x_{-1} = \infty$ . We solve difference equations for coefficients in series expansions near singular points according to the algorithm described in section 2. The corresponding results are encoded in a file present in the package:

```
{L, cis, cisrule} = Get["Data/BoundaryConditions"];
```

Here  $\mathbf{L} = L$  is a constant matrix (see section 2) and the list `cis` =  $\mathbf{c}$  defines the required information about the primary master integrals. These two quantities determine the column of constants  $\mathbf{J}_0$  in eq. (2.2) as  $\mathbf{J}_0 = L\mathbf{c}$ . The list `cis` has the form  $\{\text{ci}[1, \epsilon - 1, 0], \text{ci}[2, 0, 0], \text{ci}[3, \epsilon - 1, 0], \text{ci}[4, 0, 0]\}$ , where `ci`[ $j, n, k$ ] denotes the coefficient in front of  $x^n \ln^k x$  in  $j$ -th primary master integral. The list of replacement rules `cisrule` contains this required information which was obtained using different techniques, in particular, Mellin-Barnes representations.

The matching procedure described in the previous section is performed in our example as follows. The variable changes corresponding to the singular points are  $f_0 = x/(2 - x), f_1 = (x - 1)/(1 + 7x/9), f_2 = (9 - x)/(7 + x), f_3 = -9/(2x - 9)$ . For example, the first function maps 0 to 0, 1 to 1 and infinity to  $-1$ . In new coordinates the radius of convergence is equal to 1, however, the convergence is very slow when approaching the boarder of the convergence domain.

---

<sup>9</sup>In our paper, we use `FIRE` [36–38] in combination with `LiteRed` [39, 40].

<sup>10</sup>These explanations are also printed when the command is executed.

For adjacent regions  $i$  and  $i + 1$  we search the best possible matching point which is such  $x$  that it lies between  $x_i$  and  $x_{i+1}$  and that  $|f_i(x)| = |f_{i+1}(x)|$ . In our case we result in matching points  $\{-3, 3(3 - 2\sqrt{2}), 3, 3(3 + 2\sqrt{2})\}$ .

The matching points are separating the singular points. We have

$$-\infty < -3 < 0 < 3(3 - 2\sqrt{2}) < 1 < 3 < 9 < 3(3 + 2\sqrt{2}) < \infty.$$

Now to obtain the values in a region different from  $(-3, 3(3 - 2\sqrt{2}))$  we have to perform matching by moving in the positive or negative direction. The regions  $(3(3 - 2\sqrt{2}), 3)$  and  $(3(3 + 2\sqrt{2}), -3)$  (around infinity) are adjacent and one matching is enough. For the remaining  $(3, 3(3 + 2\sqrt{2}))$  region one performs two matchings. This procedure is performed automatically in the `Mathematica` code `DESS.m` we provide. The basic function is

```
DESS[rdatas, x, x0, oe, np]
```

It builds the evolution operator near a given point  $x_0$ , where `oe` is the order in  $\epsilon$ , `np` is the required precision, and the file `rdatas` constructed with the help of the algorithm described in section 2 contains all the required information about coefficients in expansions at all the singular points in a special format.

The action of this procedure is performed with the help of the following auxiliary functions:

```
FindLFT[x, {k,l,m}]
```

finds the Möbius transformation in  $x$  which maps  $k, l, m$  to  $(-1, 0, 1)$ ;

```
FindMPoint[x, {f1, f2}]
```

finds the matching point  $x$  with  $|f_1(x)| = |f_2(x)|$  such that  $x$  is between  $f_1^{-1}(0)$  and  $f_2^{-1}(0)$ ;

```
InverseLFT[x, f]
```

returns the inverse linear transformation function.

The manipulations with series expansion are performed in the auxiliary basis  $\mathbf{J}$ , rather in the primary basis  $\mathbf{J}_1$ . For the evaluation of  $\mathbf{J}_1$ , one takes into account the relation between the bases and applies the command

```
(T /. x -> x0).DESS[rdatas, x, x0, oe, np].L.(cis /. cisrule)
```

to evaluate the set of the primary master integrals (4.2) at the point  $x_0$  (different from the singular points) in an  $\epsilon$ -expansion up to the order `oe` with the accuracy `np`.

To test our code we ran our procedure with `oe = 15` and `np = 75` at the sample points  $-10, -3/2, 1/3, 2/3, 2, 4, 12, 25$  which lie between the singular and matching points and confirmed our results with the code `FIESTA` [41]. For example, at  $x_0 = 25$ , we obtain the following result (shown with a truncation to 10 digits) for the first primary integral:

$$\begin{aligned} &-\frac{0.25}{\epsilon^4} + \frac{2.125}{\epsilon^3} - \frac{0.2391337000}{\epsilon^2} - \frac{5.2663306926}{\epsilon} - 185.9464179437 + 6.5261388472 i \\ &-(1825.1476432369 - 48.9550593728 i)\epsilon - (8406.8551978029 - 176.0638485153 i)\epsilon^2 \\ &-(58330.4283767260 - 401.9617475893 i)\epsilon^3. \end{aligned}$$

The  $\epsilon$ -expansions here are implied with pulling out the standard factor  $e^{-\gamma_E \epsilon}$  per loop where  $\gamma_E$  is the Euler's constant.

In fact, the maximal order of expansion in  $\epsilon$  and the maximal accuracy is determined by the boundary conditions where expansion of boundary vacuum integrals is included up to  $\epsilon^3$  with the accuracy of 500 digits. This results in an  $\epsilon$ -expansion up to  $\epsilon^3$  of our primary master integrals. However, we recommend to set `oe = 15` because high negative powers of  $\epsilon$  appear in calculations. Moreover, we recommend to add the value 25 to the desired precision `np`, for a similar reason.

One more command of our code is denoted in the same way but has one more argument: `DESS[rdatas, x, f(x), oe, np, nt]`. It can be used to obtain a required number `nt` of terms of expansion near a given singular point, i.e.  $x_0 \in \{0, 1, 9, \infty\}$ . For the three finite singular points, one can request an expansion in powers of  $f(x)$  which can be any function of the form  $f(x) = (x - x_0)/(ax + b)$  different from constant, in particular,  $f(x)$  can be  $\pm(x - x_0)$ . For example, at  $x_0 = 9$ , the choice  $f(x) = 9 - x$  looks natural and leads to an expansion of the primary master integrals in powers and logarithms of  $9 - x$  with real coefficients. For the singular point  $x_0 = \infty$ , one can choose  $f(x) = (ax + b)^{-1}$ , e.g.,  $\pm 1/x$ . Here also the choice  $-1/x$  is natural for the same reason as above.

The output of this command (with one more argument) is in the form of a set of replacements  $n, j, k \rightarrow \dots$  which give results for the coefficients  $C(n + \epsilon j, k)$  in eq. (2.4) in the expansion of the evolution operator near a singular point. A result in the form of eq. (2.4) can be obtained from this result by applying the command

```
FromCoefficientRules[... , {x, x^[Epsilon], Log[x]}]
```

Similarly to the evaluation of the master integrals at a given non-singular point, one has to multiply `DESS[rdatas, x, f(x), oe, np, nt]` by `L.(cis /. cisrule)` from the right and by an expansion of the transformation matrix  $T$  near  $x = x_0$  from the left and then reexpand the product of all the factors at  $x = x_0$ .

The performance of the code depends essentially on the desired accuracy. Of course, the time needed for the evaluation with 500 digits is much more than with 50 digits because the matching procedure in the case of a high accuracy requires much more terms of expansions at singular points. For example, the evaluation at  $x = 9$  with the accuracy of 500 digits takes around 40 minutes on a laptop. However, we did not try to optimize the procedure very much. In particular, one can think of parallelization and/or putting it on `c++`.

The results for the evolution operator given by `DESS[rdatas, x, f(x), oe, np, nt]` are linear combinations of powers of  $(\pm(x - x_0))^{n+j\epsilon}$  so that it is possible to select contributions for specific  $j$  at this level. For example, one can arrive at results for the naive part of the expansion of the primary master integrals near a given finite singular point by selecting only integer powers. In fact, near  $x_0 = 0$  and  $x_0 = 1$ , we have only Taylor expansions of the master integrals in our example. We have exponents  $x^{j\epsilon}$ , with  $j=1,2,3,4$ , in the expansion at infinity but this does not mean that there is no naive expansion. The point is that the limit  $x \rightarrow \infty$  corresponds to the limit, where  $m^2 \ll |p^2|$ , so that the naive expansion in this limit reduces to the expansion of integrands in Taylor series in  $m^2$ . If one is oriented at this very limit it is reasonable to introduce a dimensionless variable in another way, as  $x = m^2/p^2$ , and then the naive expansion will be in integer powers of this variable.

## 5 Conclusion

We have presented an algorithm for the numerical evaluation of a set of master integrals depending nontrivially on one variable at a given real point with a required accuracy. The algorithm is oriented at situations where canonical form of the DE is impossible. We have provided a computer implementation of the algorithm in a simple example. This code is similar in spirit to the well-known existing codes to evaluate harmonic polylogarithms [15] and multiple polylogarithms [16], where the problem of evaluation reduces to summing up appropriate series.

We hope that one can use our algorithm and implement it to evaluate master integrals in situations where an analytic evaluation is problematic. In fact, we have provided more than the code for the evaluation of the four master integrals we considered because our package includes tools for a decomposition of the real axis into domains, a subsequent mapping and an introduction of appropriate new variables. We are thinking of a more general package which would include an automation of as many steps of the presented algorithm as possible. Input data of this package would be a matrix in DE in the normalized Fuchsian form (defined near eq. (2.5)). Output data would be the evolution operator in an epsilon expansion up to a required order with a required accuracy. In addition to the existing tools, the future package needs at least an implementation of the algorithm of section 2 to solve difference equations for series expansions at the singular points.

Of course, one can hardly construct a general algorithm to fix boundary conditions because, usually, the choice of the corresponding point and the way to obtain data for the boundary conditions is done in every situation in a special way. Still we can suggest a format for including information about the boundary conditions for using it in our future package. Anyway, our future package would check if a given system of DE is already in a global normalized Fuchsian form, with singularities on the real axis, and, if this is true, the package would automatically construct the evolution operator in an expansion up to a required order.

We discussed the problem of evaluation of Feynman integrals with two scales, i.e. dependent on one variable,  $x$ . However, one can apply DE even in the case of one-scale integrals by introducing an extra scale, solving DE with the respect to the ratio of the two scales,  $x$ , and then picking a contribution to the expansion at the point where  $x$  tends to its original value [42]. The second form of the call of `DESS` allows one to find the coefficients of the expansion of the primary master integrals near a given singular point  $x_0$ . Then it is easy to separate the ‘naive’ part of the expansion, i.e. the contribution of the non-negative integer powers of  $x - x_0$  and to find the ‘naive’ values of the primary integrals at  $x = x_0$ . For example, for the integrals considered in the previous section, this procedure can provide naive values at  $x = 1$ , i.e. integrals considered from the scratch with  $p^2$  set to  $m^2$  which are nothing but typical integrals appearing in the evaluation of the  $g - 2$  factor.

## Acknowledgments

V.S. is grateful to Oleg Veretin for useful discussions. The work of A.S. and V.S. was supported by RFBR, grant 17-02-00175A. The work of R. Lee was supported by the grant of the “Basis” foundation for theoretical physics.

**Open Access.** This article is distributed under the terms of the Creative Commons Attribution License ([CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited.

## References

- [1] A.V. Kotikov, *Differential equations method: New technique for massive Feynman diagrams calculation*, *Phys. Lett. B* **254** (1991) 158 [[INSPIRE](#)].
- [2] A.V. Kotikov, *Differential equation method: the calculation of  $N$  point Feynman diagrams*, *Phys. Lett. B* **267** (1991) 123 [Erratum *ibid.* **B 295** (1992) 409] [[INSPIRE](#)].
- [3] E. Remiddi, *Differential equations for Feynman graph amplitudes*, *Nuovo Cim. A* **110** (1997) 1435 [[hep-th/9711188](#)] [[INSPIRE](#)].
- [4] T. Gehrmann and E. Remiddi, *Differential equations for two loop four point functions*, *Nucl. Phys. B* **580** (2000) 485 [[hep-ph/9912329](#)] [[INSPIRE](#)].
- [5] T. Gehrmann and E. Remiddi, *Two loop master integrals for  $\gamma^* \rightarrow 3$  jets: the planar topologies*, *Nucl. Phys. B* **601** (2001) 248 [[hep-ph/0008287](#)] [[INSPIRE](#)].
- [6] T. Gehrmann and E. Remiddi, *Two loop master integrals for  $\gamma^* \rightarrow 3$  jets: the nonplanar topologies*, *Nucl. Phys. B* **601** (2001) 287 [[hep-ph/0101124](#)] [[INSPIRE](#)].
- [7] J.M. Henn, *Multiloop integrals in dimensional regularization made simple*, *Phys. Rev. Lett.* **110** (2013) 251601 [[arXiv:1304.1806](#)] [[INSPIRE](#)].
- [8] K.G. Chetyrkin and F.V. Tkachov, *Integration by parts: the algorithm to calculate  $\beta$ -functions in 4 loops*, *Nucl. Phys. B* **192** (1981) 159 [[INSPIRE](#)].
- [9] R.N. Lee, *Reducing differential equations for multiloop master integrals*, *JHEP* **04** (2015) 108 [[arXiv:1411.0911](#)] [[INSPIRE](#)].
- [10] O. Gituliar and V. Magerya, *Fuchsia and master integrals for splitting functions from differential equations in QCD*, *PoS(LL2016)030* [[arXiv:1607.00759](#)] [[INSPIRE](#)].
- [11] O. Gituliar and V. Magerya, *Fuchsia: a tool for reducing differential equations for Feynman master integrals to epsilon form*, *Comput. Phys. Commun.* **219** (2017) 329 [[arXiv:1701.04269](#)] [[INSPIRE](#)].
- [12] M. Prausa, *epsilon: a tool to find a canonical basis of master integrals*, *Comput. Phys. Commun.* **219** (2017) 361 [[arXiv:1701.00725](#)] [[INSPIRE](#)].
- [13] C. Meyer, *Transforming differential equations of multi-loop Feynman integrals into canonical form*, *JHEP* **04** (2017) 006 [[arXiv:1611.01087](#)] [[INSPIRE](#)].
- [14] C. Meyer, *Algorithmic transformation of multi-loop master integrals to a canonical basis with CANONICA*, *Comput. Phys. Commun.* **222** (2018) 295 [[arXiv:1705.06252](#)] [[INSPIRE](#)].
- [15] E. Remiddi and J.A.M. Vermaseren, *Harmonic polylogarithms*, *Int. J. Mod. Phys. A* **15** (2000) 725 [[hep-ph/9905237](#)] [[INSPIRE](#)].
- [16] A.B. Goncharov, *Multiple polylogarithms, cyclotomy and modular complexes*, *Math. Res. Lett.* **5** (1998) 497 [[arXiv:1105.2076](#)] [[INSPIRE](#)].
- [17] D. Maître, *HPL, a Mathematica implementation of the harmonic polylogarithms*, *Comput. Phys. Commun.* **174** (2006) 222 [[hep-ph/0507152](#)] [[INSPIRE](#)].

- [18] J. Vollinga and S. Weinzierl, *Numerical evaluation of multiple polylogarithms*, *Comput. Phys. Commun.* **167** (2005) 177 [[hep-ph/0410259](#)] [[INSPIRE](#)].
- [19] C.W. Bauer, A. Frink and R. Kreckel, *Introduction to the GiNaC framework for symbolic computation within the C++ programming language*, *J. Symb. Comput.* **33** (2000) 1 [[cs/0004015](#)] [[INSPIRE](#)].
- [20] R.N. Lee and A.A. Pomeransky, *Normalized Fuchsian form on Riemann sphere and differential equations for multiloop integrals*, [arXiv:1707.07856](#) [[INSPIRE](#)].
- [21] U. Aglietti, R. Bonciani, L. Grassi and E. Remiddi, *The two loop crossed ladder vertex diagram with two massive exchanges*, *Nucl. Phys. B* **789** (2008) 45 [[arXiv:0705.2616](#)] [[INSPIRE](#)].
- [22] R. Bonciani et al., *Two-loop planar master integrals for Higgs  $\rightarrow$  3 partons with full heavy-quark mass dependence*, *JHEP* **12** (2016) 096 [[arXiv:1609.06685](#)] [[INSPIRE](#)].
- [23] A. Primo and L. Tancredi, *On the maximal cut of Feynman integrals and the solution of their differential equations*, *Nucl. Phys. B* **916** (2017) 94 [[arXiv:1610.08397](#)] [[INSPIRE](#)].
- [24] A. Primo and L. Tancredi, *Maximal cuts and differential equations for Feynman integrals. An application to the three-loop massive banana graph*, *Nucl. Phys. B* **921** (2017) 316 [[arXiv:1704.05465](#)] [[INSPIRE](#)].
- [25] L. Adams, C. Bogner, A. Schweitzer and S. Weinzierl, *The kite integral to all orders in terms of elliptic polylogarithms*, *J. Math. Phys.* **57** (2016) 122302 [[arXiv:1607.01571](#)] [[INSPIRE](#)].
- [26] L. Adams and S. Weinzierl, *Feynman integrals and iterated integrals of modular forms*, [arXiv:1704.08895](#) [[INSPIRE](#)].
- [27] E. Remiddi and L. Tancredi, *An elliptic generalization of multiple polylogarithms*, *Nucl. Phys. B* **925** (2017) 212 [[arXiv:1709.03622](#)] [[INSPIRE](#)].
- [28] M. Czakon, *Tops from light quarks: full mass dependence at two-loops in QCD*, *Phys. Lett. B* **664** (2008) 307 [[arXiv:0803.1400](#)] [[INSPIRE](#)].
- [29] P. Bärnreuther, M. Czakon and P. Fiedler, *Virtual amplitudes and threshold behaviour of hadronic top-quark pair-production cross sections*, *JHEP* **02** (2014) 078 [[arXiv:1312.6279](#)] [[INSPIRE](#)].
- [30] S. Pozzorini and E. Remiddi, *Precise numerical evaluation of the two loop sunrise graph master integrals in the equal mass case*, *Comput. Phys. Commun.* **175** (2006) 381 [[hep-ph/0505041](#)] [[INSPIRE](#)].
- [31] B.A. Kniehl, A.F. Pikelner and O.L. Veretin, *Three-loop massive tadpoles and polylogarithms through weight six*, *JHEP* **08** (2017) 024 [[arXiv:1705.05136](#)] [[INSPIRE](#)].
- [32] R. Mueller and D.G. Öztürk, *On the computation of finite bottom-quark mass effects in Higgs boson production*, *JHEP* **08** (2016) 055 [[arXiv:1512.08570](#)] [[INSPIRE](#)].
- [33] 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](#)] [[INSPIRE](#)].
- [34] W. Wasow, *Asymptotic expansions for ordinary differential equations*, John Wiley & Sons, Inc., New York U.S.A. (1965).
- [35] K. Melnikov, L. Tancredi and C. Wever, *Two-loop  $gg \rightarrow Hg$  amplitude mediated by a nearly massless quark*, *JHEP* **11** (2016) 104 [[arXiv:1610.03747](#)] [[INSPIRE](#)].

- [36] A.V. Smirnov, *Algorithm FIRE — Feynman Integral REduction*, *JHEP* **10** (2008) 107 [[arXiv:0807.3243](#)] [[INSPIRE](#)].
- [37] A.V. Smirnov and V.A. Smirnov, *FIRE4, LiteRed and accompanying tools to solve integration by parts relations*, *Comput. Phys. Commun.* **184** (2013) 2820 [[arXiv:1302.5885](#)] [[INSPIRE](#)].
- [38] A.V. Smirnov, *FIRE5: a C++ implementation of Feynman Integral REduction*, *Comput. Phys. Commun.* **189** (2015) 182 [[arXiv:1408.2372](#)] [[INSPIRE](#)].
- [39] R.N. Lee, *Presenting LiteRed: a tool for the Loop InTEgrals REDuction*, [arXiv:1212.2685](#) [[INSPIRE](#)].
- [40] R.N. Lee, *LiteRed 1.4: a powerful tool for reduction of multiloop integrals*, *J. Phys. Conf. Ser.* **523** (2014) 012059 [[arXiv:1310.1145](#)] [[INSPIRE](#)].
- [41] A.V. Smirnov, *FIESTA4: optimized Feynman integral calculations with GPU support*, *Comput. Phys. Commun.* **204** (2016) 189 [[arXiv:1511.03614](#)] [[INSPIRE](#)].
- [42] J.M. Henn, A.V. Smirnov and V.A. Smirnov, *Evaluating single-scale and/or non-planar diagrams by differential equations*, *JHEP* **03** (2014) 088 [[arXiv:1312.2588](#)] [[INSPIRE](#)].