**AMFlow**: a Mathematica package for Feynman integrals computation via Auxiliary Mass Flow

Xiao Liu\textsuperscript{a,b,*}, Yan-Qing Ma\textsuperscript{a,c,d}

\textsuperscript{a}School of Physics and State Key Laboratory of Nuclear Physics and Technology, Peking University, Beijing 100871, China
\textsuperscript{b}Rudolf Peierls Centre for Theoretical Physics, Clarendon Laboratory, Parks Road, Oxford OX1 3PU, UK
\textsuperscript{c}Center for High Energy Physics, Peking University, Beijing 100871, China
\textsuperscript{d}Collaborative Innovation Center of Quantum Matter, Beijing 100871, China

**Abstract**

\textbf{AMFlow} is a Mathematica package to numerically compute dimensionally regularized Feynman integrals via the recently proposed auxiliary mass flow method. In this framework, integrals are treated as functions of an auxiliary mass parameter and their results can be obtained by constructing and solving linear differential systems with respect to this parameter, in an automatic way. The usage of this package is described in detail through an explicit example of double-box family involved in two-loop $t\bar{t}$ hadroproduction.

**Keywords:** Feynman integrals; Numerical evaluation; Differential equations.

*Corresponding author.

\textit{Email addresses:} xiao.liu@physics.ox.ac.uk (Xiao Liu), yqma@pku.edu.cn (Yan-Qing Ma)
PROGRAM SUMMARY

Program title: AMFlow
Developer’s repository link: https://gitlab.com/multiloop-pku/amflow
Licensing provisions: MIT
Programming language: Wolfram Mathematica 11.3 or higher
External routines/libraries used: Wolfram Mathematica [1], FiniteFlow [2], LiteRed [3], Kira [4], FIRE [5]
Nature of problem: Automatically obtaining high-precision numerical results for dimensionally regularized Feynman integrals at arbitrary points in phase-space.
Solution method: The program implements recently proposed auxiliary mass flow method, which introduces an auxiliary mass parameter to Feynman integrals and solves differential equations with respect to this parameter to obtain physical results.
Restrictions: the CPU time and the available RAM
References:
[1] http://www.wolfram.com/mathematica commercial algebraic software;
[2] https://github.com/peraro/finiteflow open source;
[3] http://www.inp.nsk.su/~lee/programs/LiteRed open source;
[4] https://gitlab.com/kira-pyred/kira open source;
[5] https://bitbucket.org/feynmanIntegrals/fire open source.
1. Introduction

Computation of Feynman integrals is crucial for the purpose of testing the standard model of particle physics and probing new physics. Currently, the main strategy is to first reduce all Feynman integrals in a problem to a small set of bases \([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]\), called master integrals, and then calculate these master integrals.

There are many methods on the market to compute master integrals, such as: sector decomposition \([22, 23, 24, 25, 26, 27, 28]\); Mellin-Barnes representation \([29, 30, 31, 32, 33, 34]\); difference equations \([2, 35]\); traditional differential equations \([36, 37, 38, 39, 40, 41, 42, 43, 44, 45]\), by setting up and solving differential equations satisfied by master integrals with respect to kinematic variables \(\vec{s}\); and others \([46, 47, 48, 49, 50, 51, 52, 53, 54, 55]\). The sector decomposition method and Mellin-Barnes representation method can be applied in principle to any integral. However, it is well known that these methods, which need to calculate multidimensional integrations directly, are very inefficient to obtain high-precision results. Difference equations and differential equations can be very efficient, but they depend on integrals reduction to set up relevant equations, which may become very nontrivial for multiloop multiscale problems. Besides, usually there is no systematic way to obtain boundary conditions for these two methods.

The auxiliary mass flow method \([56, 57]\) is also a kind of differential equations method, which calculates Feynman integrals by setting up and solving differential equations with respect to an auxiliary mass term \(\eta\). This method has many advantages. First, it is systematic, because boundary conditions at \(\eta \to \infty\) can be obtained iteratively \([57, 58]\). Second, as only ordinary differential equations are involved, high-precision results can be efficiently obtained \([59]\). Third, integrals containing linear propagators and phase-space integrations can all be calculated \([60, 61]\). Finally, integrals reduction to set up differential equations with respective to \(\eta\) is usually easier than to set up differential equations with respective to \(\vec{s}\) \([57]\). Therefore, as long as reduction tools are powerful enough to set up differential equations with respect to \(\eta\), auxiliary mass flow can always provide high-precision result efficiently.

Auxiliary mass flow method has already been used to study many physical processes \([62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76]\). Especially, equipped with the iterative strategy \([57]\), this method becomes extremely powerful so that many Feynman integrals in cutting-edge problems, which are very challenging for other methods, can be calculated (see Fig. 4 in
Ref. [57]). It is thus valuable for high-precision phenomenological studies. This paper aims to provide a public implementation of this method, including the automation of the fully iterative strategy and a high-performance numerical solver for ordinary differential equations, so that it can be more widely used for phenomenological studies.

2. Auxiliary mass flow

In this section we give a review of the auxiliary mass flow method, concentrating on the computation of normal loop integrals [56, 57]. The extensions to compute integrals containing linear propagators or phase-space integrations can be found in Refs. [60, 61].

2.1. The plain method

Let us consider a dimensionally regularized Feynman integral family defined by

\[ I(\vec{\nu}, \vec{s}, \epsilon) = \int L \prod_{i=1}^{D} d^{D} \ell_{i} \prod_{i=0}^{K+1} \frac{D_{K+1}^{-\nu_{K+1}} \cdots D_{N}^{-\nu_{N}}}{(D_{1} + i0^{+})^{\nu_{1}} \cdots (D_{K} + i0^{+})^{\nu_{K}}} \] (1)

where \( \vec{s} \) is the list of all kinematic variables including Mandelstam variables and nonzero masses of particles, \( D = 4 - 2\epsilon \) is the spacetime dimension, \( L \) is the number of loops, \( \ell_{i} \) are loop momenta, \( D_{1}, \ldots, D_{K} \) are inverse propagators, \( D_{K+1}, \ldots, D_{N} \) are irreducible scalar products introduced for completeness, \( \nu_{1}, \ldots, \nu_{K} \) can be any integers, and \( \nu_{K+1}, \ldots, \nu_{N} \) can only be non-positive integers. We next introduce an auxiliary integral family by inserting an auxiliary parameter \( \eta \) to each propagator of (1)

\[ I_{\text{aux}}(\vec{\nu}, \vec{s}, \epsilon, \eta) = \int L \prod_{i=1}^{D} d^{D} \ell_{i} \prod_{i=0}^{K+1} \frac{D_{K+1}^{-\nu_{K+1}} \cdots D_{N}^{-\nu_{N}}}{(D_{1} - \eta)^{\nu_{1}} \cdots (D_{K} - \eta)^{\nu_{K}}} \] (2)

Then physical results can be recovered by taking the following limit

\[ I(\vec{\nu}, \vec{s}, \epsilon) = \lim_{\eta \to i0^{-}} I_{\text{aux}}(\vec{\nu}, \vec{s}, \epsilon, \eta). \] (3)

This auxiliary family, although seems to be more complicated than the original one, becomes rather simple as \( \eta \) approaches the infinity. This can be understood through region analysis [77, 78]. More specifically, when \(|\eta|\) is
very large, only the integration region with \( \ell_i^\mu \sim \mathcal{O}(\sqrt{\eta}) \) can contribute, and thus every propagator can be expanded like

\[
\frac{1}{((\ell + p)^2 - m^2 - \eta)^\nu} = \frac{1}{(\ell^2 - \eta)^\nu} \sum_{i=0}^{\infty} \frac{(\nu)_i}{i!} \left( -\frac{2\ell \cdot p + p^2 - m^2}{\ell^2 - \eta} \right)^i, \tag{4}
\]

where \((\nu)_i \equiv \Gamma(\nu + i)/\Gamma(\nu)\) is the Pochhammer symbol. After all such kinds of expansion, what we get are combinations of equal-mass vacuum integrals, which have been intensively studied in literature \[79, 80, 81, 82, 83, 84\]. As a result, auxiliary integrals \( I_{\text{aux}}(\vec{\nu}, \vec{s}, \epsilon, \eta) \) in the neighborhood of \( \eta = \infty \) can be easily obtained and what remains is to perform analytic continuation (auxiliary mass flow) of them to recover physical results.

As auxiliary integrals can be expressed as linear combination of master integrals using integrals reduction, we only need to perform analytic continuation for master integrals, denoted by the vector \( \vec{I}_{\text{aux}}(\vec{s}, \epsilon, \eta) \). Integrals reduction can also setup differential equations for master integrals, which look like

\[
\frac{\partial}{\partial \eta} \vec{I}_{\text{aux}}(\vec{s}, \epsilon, \eta) = A(\epsilon, \eta) \vec{I}_{\text{aux}}(\vec{s}, \epsilon, \eta). \tag{5}
\]

For any fixed generic kinematic configuration \( \vec{s} = \vec{s}_0 \)\(^1\) the above differential equations can be numerically solved by using series expansions, similar to numerically solving differential equations with respective to kinematic variables \[85, 86\], which can realize the flow of \( \eta \) from the boundary at \( \infty \) to physical value at \( i0^- \).

Before describing how to solve the above differential equations, it is helpful to know some basic features of these auxiliary integrals as analytic functions of \( \eta \). According to Cutkosky rules \[87\], integrals can be only real-valued on the real axis when \( \eta > \eta_{\text{th}} \), where \( \eta_{\text{th}} \) is the largest threshold for the corresponding process. Thus the branch cut of the auxiliary integral can be defined as the straight line connecting \( \eta = -\infty \) and \( \eta = \eta_{\text{th}} \) along the real axis, such that the Schwarz reflection principle

\[
I_{\text{aux}}(\vec{\nu}, \vec{s}, \epsilon, \eta^*) = I_{\text{aux}}^*(\vec{\nu}, \vec{s}, \epsilon, \eta) \tag{6}
\]

\(^1\)If possible, \( \vec{s}_0 \) should be chosen as some simple rational numbers (e.g., if one only wants to obtain boundary conditions for a system of differential equations). Because the computational cost of construction of differential equations, which is usually the dominant part for complicated cutting-edge problems, depends heavily on the choice of \( \vec{s}_0 \).
holds everywhere except the branch cut (for real $s$ and $\epsilon$).

Now we can describe our strategy for analytic continuations, or solving differential equations. We first need to define a path for the analytic continuations connecting $\eta = \infty$ and $\eta = i0^-$, characterized by a list of regular points $\{\eta_0, \eta_1, \ldots, \eta_l\}$ on which we will perform series expansions in order. A typical choice is shown in Fig. 1, where the larger (smaller) circle is defined as smallest (largest) circle centered at $\eta = 0$ that contains all singularities (no singularity) except $\eta = \infty(\eta = 0)$. The choice of the regular points should satisfy the following rules: i) $\eta_0$ is outside of the larger circle; ii) $\eta_l$ is inside the smaller circle; iii) the distance between $\eta_{i+1}$ and $\eta_i$ is smaller than the convergence radius of the series expansions centered at $\eta_i$.

Then the flow of auxiliary mass can be divided into three main stages: i) expanding the integrals around $\eta = \infty$ and estimating at $\eta = \eta_0$; ii) expanding at $\eta = \eta_i$ and estimating at $\eta = \eta_{i+1}$ for $i = 0, \ldots, l - 1$; iii) expanding formally at $\eta = 0$ and matching at $\eta = \eta_l$ to determine the unknown coefficients in the formal asymptotic series. After these steps, we are able to take the limit $\eta \to i0^-$ for the expansion at $\eta = 0$ to obtain physical results.

A simple example would be helpful to explain the basic ideas of performing the expansions. For more technical details, see e.g. Refs. [56, 59]. Let us consider a massless one-loop two-point integral family

$$I(\nu_1, \nu_2, \epsilon) = \int \frac{d^D \ell}{i\pi^{D/2}} \frac{1}{((\ell^2 + i0)^{\nu_1}(\ell + p)^2 + i0)^{\nu_2}} \bigg|_{\mu^2 = 1}. \quad (7)$$

There is one master integral $I(1, 1, \epsilon)$, whose result is

$$I(1, 1, \epsilon) = (-1 - i0)^{-\epsilon} \times \frac{\Gamma(\epsilon)\Gamma(1 - \epsilon)^2}{\Gamma(2 - \epsilon)} = \frac{1}{\epsilon} + 1.42278 + 3.14159i + \mathcal{O}(\epsilon). \quad (8)$$

In the aforementioned auxiliary mass flow method, we first introduce the auxiliary mass parameter to obtain

$$I_{\text{aux}}(\nu_1, \nu_2, \epsilon, \eta) = \int \frac{d^D \ell}{i\pi^{D/2}} \frac{1}{((\ell^2 - \eta)^{\nu_1}(\ell + p)^2 - \eta)^{\nu_2}} \bigg|_{\mu^2 = 1}. \quad (9)$$

Now there are two master integrals,

$$\tilde{I}_{\text{aux}}(\epsilon, \eta) = \{I_{\text{aux}}(1, 0, \epsilon, \eta), I_{\text{aux}}(1, 1, \epsilon, \eta)\}, \quad (10)$$
Figure 1: Singularities and analytic continuations. Singularities are labeled as crosses. Solid dots are points where to perform series expansions.
and the differential equations for them are
\[
\frac{\partial}{\partial \eta} \vec{I}_{\text{aux}}(\epsilon, \eta) = \left( \begin{array}{cc}
\frac{1-\epsilon}{2(\epsilon-1)} & 0 \\
\frac{\eta}{2(\epsilon-1)} & -\frac{2(\epsilon-1)}{4\eta-1}
\end{array} \right) \vec{I}_{\text{aux}}(\epsilon, \eta).
\] (11)

Boundary condition for the first master integral can be computed fully analytically
\[
I_{\text{aux}}(1,0,\epsilon,\eta) = \eta^{1-\epsilon} \times (-\Gamma(\epsilon - 1)),
\] (12)
and the second one can be expanded near \( \eta = \infty \) giving
\[
I_{\text{aux}}(1,1,\epsilon,\eta) = \int \frac{d^D \ell}{i\pi^{D/2}} \frac{1}{(\ell^2 - \eta)^2} + \cdots = \eta^{-\epsilon} \times (\Gamma(\epsilon) + O(\eta^{-1})).
\] (13)

Next we define the list of regular points to perform expansions. We can read directly from the differential equations (11) that the singularities are \( 0, 1/4 \) and \( \infty \). As a result, the list of regular points can be chosen as \( \eta_0 = -i/2 \), \( \eta_1 = -i/4 \) and \( \eta_2 = -i/8 \).

As the first master integral in this example has been totally solved, we just consider the second one. Near \( \eta = \infty \), this integral can be expanded like
\[
I_{\text{aux}}(1,1,\epsilon,\eta) = \eta^{-\epsilon} \sum_{n=0}^{\infty} a_n(\epsilon) \eta^{-n},
\] (14)
which is a natural generalization of its boundary condition (13). We then substitute the expansions (12) and (14) into the differential equations (11) and what comes out is a system of recurrence relations which can be used to express \( a_n(\epsilon) \) in terms of \( a_0(\epsilon) \), the boundary input determined by Eq.(13).

Some of the results are given in the following
\[
a_0(\epsilon) = \frac{1}{\epsilon} - 0.577216 + O(\epsilon),
\]
\[
a_1(\epsilon) = 0.166667 + O(\epsilon),
\]
\[
\cdots
\]
\[
a_{100}(\epsilon) = 5.49443 \times 10^{-64} + O(\epsilon),
\] (15)
all of which are real-valued, if $\epsilon$ is real. The expansion \((14)\) enables us to estimate the value of $I_{\text{aux}}(1, 1, \epsilon, \eta_0)$ through

$$I_{\text{aux}}(1, 1, \epsilon, \eta_0) \approx \eta_0^{-\epsilon} \sum_{n=0}^{100} a_n(\epsilon) \eta_0^{-n} = \frac{1}{\epsilon} + 0.0548501 + 1.88709i + O(\epsilon), \quad (16)$$

where

$$\left(-\frac{i}{2}\right)^{-\epsilon} = 1 + \left(\frac{i\pi}{2} + \log(2)\right) \epsilon + O(\epsilon^2) \quad (17)$$

has been used. Expansion near the regular point $\eta = \eta_0$ is a Taylor expansion, which looks like

$$I_{\text{aux}}(1, 1, \epsilon) = \sum_{n=0}^{\infty} b_n(\epsilon)(\eta - \eta_0)^n. \quad (18)$$

We again substitute this expansion along with the value of the first master integral into the differential equations \((11)\) and obtain a system of recurrence relations, which can be used to reduce $b_n(\epsilon)$ to $b_0(\epsilon)$, the value of $I_{\text{aux}}(1, 1, \epsilon, \eta_0)$ obtained in Eq.(16). Partial results of $b_n(\epsilon)$ are shown below

$$b_0(\epsilon) = \frac{1}{\epsilon} + 0.0548501 + 1.88709i + O(\epsilon),$$
$$b_1(\epsilon) = 0.5714 - 1.77538i + O(\epsilon),$$
$$\ldots,$$
$$b_{100}(\epsilon) = -1.29958 \times 10^{24} + 1.28029 \times 10^{26}i + O(\epsilon). \quad (19)$$

Then we can estimate $I_{\text{aux}}(1, 1, \epsilon, \eta_1)$ using the expansion near $\eta = \eta_0 \quad (18)$

$$I_{\text{aux}}(1, 1, \epsilon, \eta_1) = \frac{1}{\epsilon} + 0.609168 + 2.13174i + O(\epsilon). \quad (20)$$

Similarly, we can expand near $\eta = \eta_1$ and obtain the estimation at $\eta = \eta_2$

$$I_{\text{aux}}(1, 1, \epsilon, \eta_2) = \frac{1}{\epsilon} + 0.994236 + 2.42639i + O(\epsilon). \quad (21)$$

At the last step, we need to consider the expansion near $\eta = 0$ and match at $\eta = \eta_2$. The general form of this expansion is

$$I_{\text{aux}}(1, 1, \epsilon, \eta) = \sum_{n=0}^{\infty} c_n(\epsilon) \eta^n + \eta^{1-\epsilon} \sum_{n=0}^{\infty} d_n(\epsilon) \eta^n, \quad (22)$$
where the left part comes from the homogeneous equation and the right part comes from the inhomogeneous equation (sub-topology). By substituting the expansions (12) and (22) into the differential equations (11), we can obtain two sets of recurrence relations, which can be used to reduce all $c_n(\epsilon)$ to $c_0(\epsilon)$ and determine all $d_n(\epsilon)$ respectively. For example, we have

$$c_1(\epsilon) = 2(2\epsilon - 1)c_0(\epsilon),$$
$$c_2(\epsilon) = 2(2\epsilon - 1)(2\epsilon + 1)c_0(\epsilon),$$
$$\cdots,$$

(23)

and

$$d_0(\epsilon) = -2\Gamma(\epsilon - 1),$$
$$d_1(\epsilon) = 4\Gamma(\epsilon - 1)/(\epsilon - 2),$$
$$\cdots.$$

(24)

We find there is actually only one unknown parameter, $c_0(\epsilon)$, which can be determined through matching at $\eta = \eta_2$. By substituting the estimation at $\eta = \eta_2$ (21) and the coefficients (23) and (24) into the series expansion (22), we can solve the resulting linear equation to obtain

$$c_0(\epsilon) = \frac{1}{\epsilon} + 1.42278 + 3.14159i + O(\epsilon).$$

(25)

After computing these expansions, we can finally take the physical limit $\eta \to i0^-$ in the expansion near $\eta = 0$ (22). Note that in dimensional regularization, we have

$$\lim_{\eta \to i0^-} \eta^{a+be} = 0,$$

(26)

for any nonzero $b$. So what remains in this limit is just the leading term of the Taylor part, $c_0(\epsilon)$, i.e.,

$$I(1, 1, \epsilon) \equiv \lim_{\eta \to i0^-} I_{\text{aux}}(1, 1, \epsilon, \eta)$$
$$= c_0(\epsilon)$$
$$= \frac{1}{\epsilon} + 1.42278 + 3.14159i + O(\epsilon),$$

(27)

which agrees with the analytic result [8].
2.2. Iterative strategy

One interesting phenomenon from the previous example is that the number of master integrals increases after introducing $\eta$. As a result, it can be expected for much more complicated problems, the introduction of $\eta$ may greatly increase the number of MIs, such that the differential equations cannot be set up in reasonable time with current reduction techniques. To overcome this difficulty, in [57] we propose to apply the auxiliary mass flow method iteratively to reduce the number of master integrals, and thus the computational cost, to a reasonable level.

The key observation is that the number of master integrals can be reduced if $\eta$ is introduced to fewer propagators. For example, for the two-loop five-point massless double-pentagon integral family with 108 master integrals shown in Figure 2, we obtain 476 master integrals if $\eta$ is introduced to all propagators (“all” mode), 319 master integrals for propagators 1-6 (“loop” mode), 233 master integrals for propagators 4-6 (“branch” mode), and the best case, 176 master integrals for the propagator 5 (“propagator” mode). For topologies where independent internal masses exist, we can do even better. We can simply treat these masses as $\eta$ and thus will not introduce any extra mass scale (“mass” mode). Because “mass” and “propagator” mode introduce fewer extra number of master integrals than other modes in general, they usually perform better.

However, as an expense, the boundary analysis is more complicated in general, due to more contributing integration regions as $\eta \to \infty$. Following the general rules of region analysis [77, 78], inequivalent regions can be characterized by the size of loop momentum carried by each branch of the diagram, which can be either of $O(\sqrt{\eta})$ (denoted as large loop momentum, L) or $O(1)$ (denoted as small loop momentum, S). For example, for a two-loop
integral family with three branches, whose loop momenta can be chosen as \( \ell_1, \ell_2 \) and \( \ell_1 + \ell_2 \), at most five regions may contribute: (LLL), (LLS), (LSL), (SLL) and (SSS).

To obtain boundary conditions, we need to expand the integrands of master integrals in each region. Specifically, in the all-large region (L...L), each propagator should be expanded as

\[
\frac{1}{((\ell + p)^2 - m^2 - \kappa \eta)\nu} \sim \frac{1}{(\ell^2 - \kappa \eta)^\nu},
\]

where \( \kappa = 1 \) or 0, depending on whether \( \eta \) is introduced to this propagator or not. We thus obtain vacuum integrals in this region. In the all-small region (S...S), only propagators containing \( \eta \) should be expanded as

\[
\frac{1}{((\ell + p)^2 - m^2 - \eta)\nu} \sim \frac{1}{(-\eta)^\nu}.
\]

In this case, we get integrals in a subfamily, with propagators containing \( \eta \) contracted. In mixed regions, we need to decompose loop momentum of each propagator as the sum of a large part \( \ell_L \) and a small part \( \ell_S \). Then, if \( \ell_L \neq 0 \) or \( \kappa \neq 0 \), we can expand the propagator as

\[
\frac{1}{((\ell_L + \ell_S + p)^2 - m^2 - \kappa \eta)\nu} \sim \frac{1}{(\ell_L^2 - \kappa \eta)^\nu}.
\]

Otherwise, no expansion is needed. After the expansion, the part containing large loop momenta and the part containing small loop momenta are decoupled and we obtain factorized integrals.

It turns out that usually the boundary integrals are still too complicated to evaluate directly. But this is still fine because they are already simpler than the original integrals, which means we can keep applying the previous procedure to simplify the boundary integrals until they are all known to us. For example, the double-pentagon topology can be simplified iteratively with “propagator” mode as shown in Fig. 3.

In practice, we would profit from a systematic definition of the terminal topologies. For example, we can always identify single-mass vacuum integrals as our terminals. In Ref. [58], single-mass vacuum integrals are further simplified in an iterative manner. In that way, we can simply identify the 0-loop integral (whose result is 1) as terminals, which have been proved to be more convenient.
3. Numerical fit

A very useful trick implemented in AMFlow is numerical fit. Consider a function $f(x)$ which can be expanded near $x = 0$ as

$$f(x) = \sum_{n=0}^{\infty} f_n x^n,$$

and our goal is to compute its estimation up to the $k$-th order

$$\tilde{f}_0 + \tilde{f}_1 x + \cdots + \tilde{f}_k x^k,$$

with relative accuracy $E_n \leq E$, where $\tilde{f}_n$ is the estimation of $f_n$ and $E_n$ is defined by

$$E_n = \left| \frac{\tilde{f}_n - f_n}{f_n} \right|.$$

We propose to realize this by evaluating \( f(x) \) numerically at some sample points \( x_0, x_1, \ldots, x_N \) \((N \geq k)\) near \( x = 0 \) and solving a system of linear equations

\[
\begin{align*}
  f(x_0) &= \tilde{f}_0 + \tilde{f}_1 x_0 + \cdots + \tilde{f}_N x_0^N, \\
  f(x_1) &= \tilde{f}_0 + \tilde{f}_1 x_1 + \cdots + \tilde{f}_N x_1^N, \\
  \vdots \\
  f(x_N) &= \tilde{f}_0 + \tilde{f}_1 x_N + \cdots + \tilde{f}_N x_N^N.
\end{align*}
\] (34)

In practice, we find two ways are useful to choose these sample points:

1. \( |x_0| \sim \cdots \sim |x_N| \sim r \ll R \),
2. \( x_0, \ldots, x_N \) are distributed uniformly on the circle centered at \( x = 0 \) with radius \( r < R \),

where \( R \) is the convergence radius of the expansion (31). If one of these ways is chosen and the precision \( p \) of the samples \( f(x_0), f(x_1), \ldots, f(x_N) \) is sufficiently high, then the relative accuracy of \( f_n \) can be roughly estimated as

\[
E_n \sim \left( \frac{r}{R} \right)^{N+1-n}, \quad 0 \leq n \leq N.
\] (35)

It can be seen that the relative accuracy \( E_n \) decreases as \( n \) increases. Thus to achieve our precision goal, we can set \( E_k \sim E \), or equivalently

\[
N \sim k - 1 + \frac{\log(E)}{\log(r/R)}.
\] (36)

This also gives a constraint about the precision \( p \) of the samples

\[
p \lesssim E_0 \sim \exp(k \log(r/R) + \log(E)),
\] (37)

because we cannot expect a correct result if the precision of the samples is too low. So the total time consumption to obtain the estimation (32) is

\[
T = (N + 1) \times t(p) \gtrsim \left( k + \frac{\log(E)}{\log(r/R)} \right) \times t(\exp(k \log(r/R) + \log(E))),
\] (38)

where \( t(p) \) is the average time needed to compute at a sample point with precision \( p \), depending on both the nature of the problem and the numerical
algorithm. Typically, in the framework of power series expansion method to solve differential equations of Feynman integrals, the dominant part of \( t(p) \) is a polynomial-like object of the number of correct digits, i.e.,

\[
t(p) \sim (- \log(p))^\alpha,
\]

where \( \alpha \) is a positive number. Therefore, we have

\[
T \gtrsim \left( k + \frac{\log(E)}{\log(r/R)} \right) \times (-k \log(r/R) - \log(E))^\alpha,
\]

which can be minimized by choosing

\[
r \sim RE^{1/(\alpha k)},
\]

and

\[
N \sim (\alpha + 1)k - 1, \quad p \lesssim E^{(\alpha+1)/\alpha}.
\]

Next we can discuss how to apply this trick to the computation of Feynman integrals. To obtain numerical results of master integrals as expansions in \( \epsilon \), we can solve differential equations with some numerical values of \( \epsilon \) and solve a system of linear equations like Eq. (34) for each master integral. We find the first way to choose sample points stated after Eq. (34) is better to use in this case, because we can always choose real values of \( \epsilon \) to avoid potential complexities. Note that (41) and (42) only serve as a reference, and in practice one may need to make some adjustments to get satisfactory results.

This trick brings several benefits. First, a much simpler code structure is made possible, because in this framework all integrals are simply pure numbers rather than expansions in \( \epsilon \), which is much easier to carry out. Second, the problem of \( \epsilon \)-order cancellations is totally resolved, because we never use truncated series in \( \epsilon \) to express any integral throughout the calculations, which means we can always include more \( \epsilon \)-orders by simply increasing the precision of the integrals. Finally, the computations at different sample points are totally independent and thus can be massively parallelized to save our waiting time.

\[\text{Suppose these integrals have been normalized such that they all start with } \epsilon^0.\]
This trick can also be applied to achieve asymptotic expansions of Feynman integrals at a given phase-space point or a given value of \( \eta \). Sometimes, this becomes crucial, given the fact that there are usually many removable singularities in differential equations. With the second way to choose sample points on a circle, removable singularities inside the circle can be totally ignored.

4. Using AMFlow

The latest version of AMFlow can be downloaded from

\[ \text{https://gitlab.com/multiloop-pku/amflow} \] \hspace{1cm} (43)

Users can then follow the guidance outlined in README.md to install this package properly on their devices. After that, the package can be loaded by the command

\[
\text{Get["/path/to/AMFlow.m"]};
\]

AMFlow depends on external programs to do integrals reduction. To use different reducers, one can set the following option

\[
\text{SetReductionOptions["IBPReducer" -> reducer]};
\]

where reducer can be any reducer whose interface with AMFlow has been built. Currently, three reducers based on Laporta's algorithm are available, including "FiniteFlow+LiteRed" \[6, 14\], "FIRE+LiteRed" \[6, 17\] and "Kira" \[18\]. Other reducers can also play their roles after users build their interfaces with AMFlow properly.

The usage of AMFlow is best illustrated with an example. Fig. 4 shows a two-loop planar integral family involved in NNLO QCD corrections to \( \bar{t}t \) hadroproduction. The four external momenta \( \{p_1, p_2, p_3, p_4\} \) flowing into the diagram satisfy the momentum conservation \( p_1 + p_2 + p_3 + p_4 = 0 \) and on-shell conditions \( p_1^2 = p_2^2 = 0, p_3^2 = p_4^2 = m^2 \). Besides, there are two independent kinematic variables \( s = (p_1 + p_2)^2 \) and \( t = (p_1 + p_3)^2 \). The inverse propagators for this diagram can be written as

\[
\mathcal{D}_1 = \ell_1^2, \quad \mathcal{D}_2 = (\ell_1 + p_1)^2, \quad \mathcal{D}_3 = (\ell_1 + p_1 + p_2)^2, \quad \mathcal{D}_4 = \ell_2^2, \quad \\
\mathcal{D}_5 = (\ell_2 + p_3)^2 - m^2, \quad \mathcal{D}_6 = (\ell_2 + p_3 + p_4)^2, \quad \mathcal{D}_7 = (\ell_1 + \ell_2)^2, \quad (44)
\]

and two irreducible scalar products can be chosen as

\[
\mathcal{D}_8 = (\ell_1 - p_3)^2, \quad \mathcal{D}_9 = (\ell_2 + p_1)^2. \quad (45)
\]
Suppose our final goal is to compute the following four top-sector integrals
\[
I(1, 1, 1, 1, 1, 1, 1, -3, 0), I(1, 1, 1, 1, 1, 1, -2, -1),
I(1, 1, 1, 1, 1, 1, -1, -2), I(1, 1, 1, 1, 1, 1, 0, -3),
\] (46)
from \(\epsilon^{-4}\) to \(\epsilon^0\), with 20-digit coefficients, at a numerical kinematic point
\[
s = 30, \quad t = -10/3, \quad m^2 = 1.
\] (47)
This example can be found in examples/automatic_vs_manual/run.wl.

Figure 4: A two-loop planar integral family involved in NNLO QCD corrections to \(t\bar{t}\) hadroproduction.

4.1. Input
First, we should use the function \texttt{AMFlowInfo} to define globally used objects during the computation, like
\[
\text{AMFlowInfo}[\text{key}] = \text{obj};
\]
where \texttt{key} should be a string pre-defined in \texttt{AMFlow} and \texttt{obj} should be the corresponding object. We list most frequently used pre-defined strings and the meaning of their corresponding objects below:

- "Family" - the name of the integral family;
- "Loop" - a list of all loop momenta;
- "Leg" - a list of all external momenta;
- "Conservation" - a list of replacement rules for momentum conservation;
- "Replacement" - a list of complete replacement rules for scalar products among external legs;
- "Propagator" - a list of complete inverse propagators;
- "Numeric" - a list of replacement rules indicating the numerical kinematics where to perform the computation;
"NThread" - the number of threads in use.

In this example, we can simply write, e.g.,

\[
\text{AMFlowInfo["Family"] = tt;}
\]

\[
\text{AMFlowInfo["Loop"] = {l1, l2};}
\]

\[
\text{AMFlowInfo["Leg"] = {p1, p2, p3, p4};}
\]

\[
\text{AMFlowInfo["Conservation"] = {p4 -> -p1-p2-p3;}
\]

\[
\text{AMFlowInfo["Replacement"] = {p1^2 -> 0, p2^2 -> 0, p3^2 -> msq, p4^2 -> msq, (p1+p2)^2 -> s, (p1+p3)^2 -> t};}
\]

\[
\text{AMFlowInfo["Propagator"] = {l1^2, (l1+p1)^2, (l1+p1+p2)^2, l2^2, (l2+p3)^2-msq, (l2+p3+p4)^2, (l1+l2)^2, (l1-p3)^2, (l2+p1)^2};}
\]

\[
\text{AMFlowInfo["Numeric"] = {s -> 30, t -> -10/3, msq -> 1};}
\]

\[
\text{AMFlowInfo["NThread"] = 4;}
\]

\[4.2. \text{Automatic computation}\]

AMFlow provides a function named \texttt{SolveIntegrals} to perform automatic computations of Feynman integrals. A general usage of this function should be like

\[
\text{auto = SolveIntegrals[target, goal, epsorder];}
\]

where \texttt{target} is a list of target integrals, \texttt{goal} represents the precision goal and \texttt{epsorder} means the length of $\epsilon$ expansion in the final expansions, i.e., starting from $\epsilon^{-2L}$ and ending at $\epsilon^{-2L+\text{order}}$ with $L$ the number of loops. This function will first reduce the target integrals to master integrals and then compute master integrals using auxiliary mass flow. The output \texttt{auto} is a list of replacement rules from integrals to their values.

For current example, we can write

\[
\text{target = } \{j[tt,1,1,1,1,1,1,-3,0], j[tt,1,1,1,1,1,1,-2,-1],
\]

\[
\quad j[tt,1,1,1,1,1,1,-1,-2], j[tt,1,1,1,1,1,1,0,-3]\};
\]

\[
\text{goal } = 20;
\]

\[
\text{epsorder } = 4;
\]

where we have adopted the notation of \texttt{LiteRed} to represent an integral

\[
\text{j[tt,}\nu_1,\ldots,\nu_9] \leftrightarrow I(\nu_1,\ldots,\nu_9) \text{ in family tt. } (48)
\]

After the computation, the output \texttt{auto} should be like (if "FiniteFlow+LiteRed" is chosen)

\[
\{j[tt,1,1,1,1,1,1,1,-2,-1] \rightarrow \}
\]
\[ -0.029131054131054131054/\epsilon^4 \\
+0.15634543151250003740/\epsilon^3 \\
-(0.007823397125433531023-0.138772644276800647440I)/\epsilon^2 \\
+(6.4018478848121593013-5.3096594693278082225I)/\epsilon \\
+(6.385202185942958097+49.103186001778095122I), \\
\ldots \]

where \( \epsilon \) means the dimensional regulator \( \epsilon \).

4.3. Manual computation

Although \texttt{SolveIntegrals} is designed for most general purposes, there could be some extreme cases where this function may not be able to produce satisfactory results. So we introduce a more involved way to compute integrals in this section.

We first use the function \texttt{GenerateNumericalConfig} to regenerate the parameters for numerical evaluation suggested by \texttt{SolveIntegrals}:

\[
\{\text{epslist}, \text{workingpre}, \text{xorder}\} = \text{GenerateNumericalConfig}[\text{goal, epsorder}];
\]

where \( \text{goal} \) and \( \text{epsorder} \) have been defined in the previous section. The output is a triblet: \text{epslist} is a list of suggested sample points of \( \epsilon \), \text{workingpre} is the suggested working precision and \text{xorder} is the suggested truncated order of the power series expansions. In principle, if these suggested parameters are used, we will obtain exactly the same results as \texttt{SolveIntegrals}. So, when \texttt{SolveIntegrals} fails to generate satisfactory results, users can define their own \text{epslist}, \text{workingpre} and \text{xorder}.

We then tell the program our preferred parameters by

\[
\text{SetAMFOptions["WorkingPre"->workingpre, "XOrder"->xorder];}
\]

and compute target integrals on \( \epsilon \)-samples by

\[
\text{soleps} = \text{BlackBoxAMFlow[target, epslist];}
\]

where \text{target} is the list of target integrals defined in the previous section. The output \text{soleps} is a list of replacement rules like

\[
\{j[tt,1,1,1,1,1,-2,-1] \rightarrow \{v11,v12,\ldots,v1n\}, \\
\quad j[tt,1,1,1,1,1,-1,-2] \rightarrow \{v21,v22,\ldots,v2n\}, \ldots \}
\]

where \( vij \) are pure numbers, representing the value of the \( i \)-th integral on the \( j \)-th \( \epsilon \)-sample. After that, we need to fit the expansions in \( \epsilon \) using these samples. This can be achieved by

\[
\text{exp} = \text{FitEps[epslist, \\
\#, leading]/@Values[soleps];}
\]

where \text{leading} means the leading power of \( \epsilon \)-pole in the expansions, which
can be set to $-4$ in this example. The output `exp` is just the list of expansions in `eps` for target integrals.

4.4. Other functions

There are also other useful functions in AMFlow. Here we just give a brief summary. For more details, users can investigate corresponding examples provided in the folder `examples`.

1. Computation of integrals containing linear propagators [60].
   See `linear_propagator`.
2. Computation of phase space integrations [61].
   See `automatic_phasespace` and `feynman_prescription`.
3. Computation of asymptotic expansions using the differential equations solver provided in AMFlow, either by traditional matching or numerical fit introduced in section [3]
   See `differential_equation_solver`.
4. Computation of integrals with complex kinematic parameters.
   See `complex_kinematics`.
5. Computation of integrals in arbitrary space-time dimension.
   See `spacetime_dimension`.

4.5. Summary of options

AMFlow allows users to set global options through `SetAMFOptions`, `SetReductionOptions` and `SetReducerOptions`. Here we list and describe the most frequently used options. For other options, we refer the users to the file `options_summary`.

| Option and Default | Description |
|--------------------|-------------|
| `SetAMFOptions`    |             |
| "D0"→4            | A rational number $D_0$ such that the integrals will be computed with $D = D_0 - 2\epsilon$. |
| "WorkingPre"→100  | Working precision when performing numerical computations, including solving differential equations and fitting. |
| "XOrder"→100      | Truncated order of expansions when solving differential equations. |
| `SetReductionOptions` |               |
| Option                  | Description                                                                                                                                 |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| "IBPReducer" → "FiniteFlow+LiteRed" | Integration-by-parts reducer. Available reducers include "FiniteFlow+LiteRed", "Kira" and "FIRE+LiteRed". |
| "BlackBoxRank" → 3     | Suggested maximal rank of seed integrals when constructing IBP systems. But if the maximal rank of target integrals \( s \) is larger than the value of this option, then the maximal rank of seed integrals will be adjusted to \( s \) internally. |
| "BlackBoxDot" → 0      | Suggested maximal dot of seed integrals when constructing IBP systems. But if the maximal dot of target integrals \( r \) is larger than the value of this option, then the maximal dot of seed integrals will be adjusted to \( r \) internally. |
| "EMSymmetry" → False   | A parameter indicating whether symmetries among external legs should be exploited when preparing the topology using LiteRed. |
| "IntegralOrder" → 5    | A positive integer ranging from 1 to 8 specifying the integral ordering for Kira. For more details, see Ref. [88]. |
| "ReductionMode" → "Kira" | Reduction mode for Kira. Available modes include "Kira", "FireFly", "Mixed" and "NoFactorScan". See Ref. [18] for more details. |

5. Summary and outlook

In this paper, the Mathematica package AMFlow is presented together with some explicit examples. We have highlighted the numerical fit strategy, which can overcome many difficulties when numerically solving differential equations.

The differential equations solver provided in AMFlow is of high performance and very suitable for high precision computations. In later version of AMFlow, we will provide some functions for users to access this solver in a more convenient way.

With the auxiliary mass flow method, integral reduction will be the only input for calculating Feynman integrals [58]. In the near future, a public
implementation of the reduction method developed in Refs. [11, 12] will be available, which can typically reduce the time consumption by 2 orders of magnitude comparing with other methods on the market. With this powerful reduction package and AMFlow, complicated integrals such as those in Ref. [57], can be computed automatically.

Acknowledgements

We thank Z. F. Liu and C. Y. Wang for many useful discussions. This work was supported in part by the National Natural Science Foundation of China (Grants No. 11875071, No. 11975029), the National Key Research and Development Program of China under Contracts No. 2020YFA0406400 and the High-performance Computing Platform of Peking University. The research of XL was also supported by the ERC Starting Grant 804394 HipQCD and by the UK Science and Technology Facilities Council (STFC) under grant ST/T000864/1.

References

[1] K. G. Chetyrkin and F. V. Tkachov, Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops, Nucl. Phys. B192 (1981) 159–204 [InSPIRE].

[2] S. Laporta, High precision calculation of multiloop Feynman integrals by difference equations, Int. J. Mod. Phys. A 15 (2000) 5087–5159 [hep-ph/0102033] [InSPIRE].

[3] J. Gluza, K. Kajda, and D. A. Kosower, Towards a Basis for Planar Two-Loop Integrals, Phys. Rev. D83 (2011) 045012 [arXiv:1009.0472] [InSPIRE].

[4] R. M. Schabinger, A New Algorithm For The Generation Of Unitarity-Compatible Integration By Parts Relations, JHEP 01 (2012) 077 [arXiv:1111.4220] [InSPIRE].

[5] A. von Manteuffel and C. Studerus, Reduze 2 - Distributed Feynman Integral Reduction, [arXiv:1201.4330] [InSPIRE].

[6] 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].
[7] A. von Manteuffel and R. M. Schabinger, *A novel approach to integration by parts reduction*, Phys. Lett. B744 (2015) 101–104 [arXiv:1406.4513] [InSPIRE].

[8] K. J. Larsen and Y. Zhang, *Integration-by-parts reductions from unitarity cuts and algebraic geometry*, Phys. Rev. D93 (2016) 041701 [arXiv:1511.01071] [InSPIRE].

[9] T. Peraro, *Scattering amplitudes over finite fields and multivariate functional reconstruction*, JHEP 12 (2016) 030 [arXiv:1608.01902] [InSPIRE].

[10] P. Mastrolia and S. Mizera, *Feynman Integrals and Intersection Theory*, JHEP 02 (2019) 139 [arXiv:1810.03818] [InSPIRE].

[11] X. Liu and Y.-Q. Ma, *Determining arbitrary Feynman integrals by vacuum integrals*, Phys. Rev. D 99 (2019) 071501 [arXiv:1801.10523] [InSPIRE].

[12] X. Guan, X. Liu, and Y.-Q. Ma, *Complete reduction of two-loop five-light-parton scattering amplitudes*, Chin. Phys. C 44 (2020) 9 [arXiv:1912.09294] [InSPIRE].

[13] J. Klappert and F. Lange, *Reconstructing rational functions with FireFly*, Comput. Phys. Commun. 247 (2020) 106951 [arXiv:1904.00009] [InSPIRE].

[14] T. Peraro, *FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs*, JHEP 07 (2019) 031 [arXiv:1905.08019] [InSPIRE].

[15] H. Frellesvig, F. Gasparotto, S. Laporta, M. K. Mandal, P. Mastrolia, L. Mattiazzi, and S. Mizera, *Decomposition of Feynman Integrals on the Maximal Cut by Intersection Numbers*, JHEP 05 (2019) 153 [arXiv:1901.11510] [InSPIRE].

[16] Y. Wang, Z. Li, and N. Ul Basat, *Direct reduction of multiloop multiscale scattering amplitudes*, Phys. Rev. D 101 (2020) 076023 [arXiv:1901.09390] [InSPIRE].
[17] A. V. Smirnov and F. S. Chuharev, *FIRE6: Feynman Integral Reduction with Modular Arithmetic*, arXiv:1901.07808 [InSPIRE].

[18] J. Klappert, F. Lange, P. Maierhöfer, and J. Usovitsch, *Integral reduction with Kira 2.0 and finite field methods*, Comput. Phys. Commun. **266** (2021) 108024 [arXiv:2008.06494] [InSPIRE].

[19] J. Boehm, M. Wittmann, Z. Wu, Y. Xu, and Y. Zhang, *IBP reduction coefficients made simple*, JHEP **12** (2020) 054 [arXiv:2008.13194] [InSPIRE].

[20] M. Heller and A. von Manteuffel, *MultivariateApart: Generalized Partial Fractions*, arXiv:2101.08283 [InSPIRE].

[21] D. Bendle, J. Boehm, M. Heymann, R. Ma, M. Rahn, L. Ristau, M. Wittmann, Z. Wu, and Y. Zhang, *Two-loop five-point integration-by-parts relations in a usable form*, arXiv:2104.06866 [InSPIRE].

[22] K. Hepp, *Proof of the Bogolyubov-Parasiuk theorem on renormalization*, Commun. Math. Phys. **2** (1966) 301–326 [InSPIRE].

[23] M. Roth and A. Denner, *High-energy approximation of one loop Feynman integrals*, Nucl. Phys. B **479** (1996) 495–514 [hep-ph/9605420] [InSPIRE].

[24] T. Binoth and G. Heinrich, *An automatized algorithm to compute infrared divergent multiloop integrals*, Nucl. Phys. B **585** (2000) 741–759 [hep-ph/0004013] [InSPIRE].

[25] G. Heinrich, *Sector Decomposition*, Int. J. Mod. Phys. A **23** (2008) 1457–1486 [arXiv:0803.4177] [InSPIRE].

[26] A. V. Smirnov, *FIESTA4: Optimized Feynman integral calculations with GPU support*, Comput. Phys. Commun. **204** (2016) 189–199 [arXiv:1511.03614] [InSPIRE].

[27] S. Borowka, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk, and T. Zirke, *SecDec-3.0: numerical evaluation of multi-scale integrals beyond one loop*, Comput. Phys. Commun. **196** (2015) 470–491 [arXiv:1502.06595] [InSPIRE].
[28] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk, and T. Zirke, *pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*, *Comput. Phys. Commun.* **222** (2018) 313–326 [arXiv:1703.09692] [InSPIRE].

[29] E. E. Boos and A. I. Davydychev, *A Method of evaluating massive Feynman integrals*, *Theor. Math. Phys.* **89** (1991) 1052–1063 [InSPIRE]. [Teor. Mat. Fiz.89,56(1991)].

[30] V. A. Smirnov, *Analytical result for dimensionally regularized massless on shell double box*, *Phys. Lett. B* **460** (1999) 397–404 [hep-ph/9905323] [InSPIRE].

[31] J. B. Tausk, *Nonplanar massless two loop Feynman diagrams with four on-shell legs*, *Phys. Lett. B* **469** (1999) 225–234 [hep-ph/9909506] [InSPIRE].

[32] M. Czakon, *Automatized analytic continuation of Mellin-Barnes integrals*, *Comput. Phys. Commun.* **175** (2006) 559–571 [hep-ph/0511200] [InSPIRE].

[33] A. V. Smirnov and V. A. Smirnov, *On the Resolution of Singularities of Multiple Mellin-Barnes Integrals*, *Eur. Phys. J. C* **62** (2009) 445–449 [arXiv:0901.0386] [InSPIRE].

[34] J. Gluza, K. Kajda, and 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] [InSPIRE].

[35] R. N. Lee, *Space-time dimensionality D as complex variable: Calculating loop integrals using dimensional recurrence relation and analytical properties with respect to D*, *Nucl. Phys. B* **830** (2010) 474–492 [arXiv:0911.0252] [InSPIRE].

[36] A. V. Kotikov, *Differential equations method: New technique for massive Feynman diagrams calculation*, *Phys. Lett. B* **254** (1991) 158–164 [InSPIRE].
[37] A. V. Kotikov, *Differential equation method: The Calculation of N point Feynman diagrams*, *Phys. Lett. B* **267** (1991) 123–127 [InSPIRE]. [Erratum: Phys.Lett.B 295, 409–409 (1992)].

[38] E. Remiddi, *Differential equations for Feynman graph amplitudes*, *Nuovo Cim.* **A110** (1997) 1435–1452 [hep-th/9711188] [InSPIRE].

[39] T. Gehrmann and E. Remiddi, *Differential equations for two loop four point functions*, *Nucl. Phys.* **B580** (2000) 485–518 [hep-ph/9912329] [InSPIRE].

[40] M. Argeri and P. Mastrolia, *Feynman Diagrams and Differential Equations*, *Int. J. Mod. Phys.* **A** **22** (2007) 4375–4436 [arXiv:0707.4037] [InSPIRE].

[41] S. Müller-Stach, S. Weinzierl, and R. Zayadeh, *Picard-Fuchs equations for Feynman integrals*, *Commun. Math. Phys.* **326** (2014) 237–249 [arXiv:1212.4389] [InSPIRE].

[42] J. M. Henn, *Multiloop integrals in dimensional regularization made simple*, *Phys. Rev. Lett.* **110** (2013) 251601 [arXiv:1304.1806] [InSPIRE].

[43] J. M. Henn, *Lectures on differential equations for Feynman integrals*, *J. Phys.* **A48** (2015) 153001 [arXiv:1412.2296] [InSPIRE].

[44] F. Moriello, *Generalised power series expansions for the elliptic planar families of Higgs + jet production at two loops*, *JHEP* **01** (2020) 150 [arXiv:1907.13234] [InSPIRE].

[45] M. Hidding, *DiffExp, a Mathematica package for computing Feynman integrals in terms of one-dimensional series expansions*, *Comput. Phys. Commun.* **269** (2021) 108125 [arXiv:2006.05510] [InSPIRE].

[46] S. Catani, T. Gleisberg, F. Krauss, G. Rodrigo, and J.-C. Winter, *From loops to trees by-passing Feynman’s theorem*, *JHEP* **09** (2008) 065 [arXiv:0804.3170] [InSPIRE].

[47] G. Rodrigo, S. Catani, T. Gleisberg, F. Krauss, and J.-C. Winter, *From multileg loops to trees (by-passing Feynman’s Tree Theorem)*, *Nucl. Phys. B Proc. Suppl.* **183** (2008) 262–267 [arXiv:0807.0531] [InSPIRE].

26
[48] I. Bierenbaum, S. Catani, P. Draggiotis, and G. Rodrigo, *A Tree-Loop Duality Relation at Two Loops and Beyond*, *JHEP* **10** (2010) 073 [arXiv:1007.0194] [InSPIRE]

[49] I. Bierenbaum, S. Buchta, P. Draggiotis, I. Malamos, and G. Rodrigo, *Tree-Loop Duality Relation beyond simple poles*, *JHEP* **03** (2013) 025 [arXiv:1211.5048] [InSPIRE]

[50] E. T. Tomboulis, *Causality and Unitarity via the Tree-Loop Duality Relation*, *JHEP* **05** (2017) 148 [arXiv:1701.07052] [InSPIRE]

[51] R. Runkel, Z. Szőr, J. P. Vesga, and S. Weinzierl, *Causality and loop-tree duality at higher loops*, *Phys. Rev. Lett.* **122** (2019) 111603 [arXiv:1902.02135] [InSPIRE], [Erratum: Phys.Rev.Lett. 123, 059902 (2019)].

[52] Z. Capatti, V. Hirschi, D. Kermanschah, and B. Ruijl, *Loop-Tree Duality for Multiloop Numerical Integration*, *Phys. Rev. Lett.* **123** (2019) 151602 [arXiv:1906.06138] [InSPIRE]

[53] J. J. Aguilera-Verdugo, F. Driencourt-Mangin, R. J. Hernández-Pinto, J. Plenter, S. Ramirez-Uribe, A. E. Renteria Olivo, G. Rodrigo, G. F. R. Sborlini, W. J. Torres Bobadilla, and S. Tracz, *Open Loop Amplitudes and Causality to All Orders and Powers from the Loop-Tree Duality*, *Phys. Rev. Lett.* **124** (2020) 211602 [arXiv:2001.03564] [InSPIRE]

[54] Q. Song and A. Freitas, *On the evaluation of two-loop electroweak box diagrams for $e^+e^- \rightarrow HZ$ production*, *JHEP* **04** (2021) 179 [arXiv:2101.00308] [InSPIRE]

[55] I. Dubovyk, A. Freitas, J. Gluza, K. Grzanka, M. Hidding, and J. Usovitsch, *Evaluation of multi-loop multi-scale Feynman integrals for precision physics*, [arXiv:2201.02576] [InSPIRE]

[56] X. Liu, Y.-Q. Ma, and C.-Y. Wang, *A Systematic and Efficient Method to Compute Multi-loop Master Integrals*, *Phys. Lett.* **B779** (2018) 353–357 [arXiv:1711.09572] [InSPIRE]

[57] X. Liu and Y.-Q. Ma, *Multiloop corrections for collider processes using auxiliary mass flow*, [arXiv:2107.01864] [InSPIRE]
[58] Z.-F. Liu and Y.-Q. Ma, *Feynman integrals are completely determined by linear algebra*, arXiv:2201.11637 [InSPIRE].

[59] W. Wason, *Asymptotic Expansions for Ordinary Differential Equations*. Dover Publications, Inc., 1987 [InSPIRE].

[60] Z.-F. Liu and Y.-Q. Ma, *Automatic computation of Feynman integrals containing linear propagators via auxiliary mass flow*, Phys. Rev. D 105 (2022) 074003 arXiv:2201.11636 [InSPIRE].

[61] X. Liu, Y.-Q. Ma, W. Tao, and P. Zhang, *Calculation of Feynman loop integration and phase-space integration via auxiliary mass flow*, Chin. Phys. C 45 (2021) 031115 arXiv:2009.07987 [InSPIRE].

[62] P. Zhang, C.-Y. Wang, X. Liu, Y.-Q. Ma, C. Meng, and K.-T. Chao, *Semi-analytical calculation of gluon fragmentation into $1S_0^{[1,8]}$ quarkonia at next-to-leading order*, JHEP 04 (2019) 116 arXiv:1810.07656 [InSPIRE].

[63] Q. Yang, R.-Y. Zhang, M.-M. Long, S.-M. Wang, W.-G. Ma, J.-W. Zhu, and Y. Jiang, *QCD Corrections to $e^+e^- \to H^\pm W^\mp$ in Type-I THDM at Electron Positron Colliders*, Chin. Phys. C 44 (2020) 093101 arXiv:2005.11010 [InSPIRE].

[64] C. Brønnum-Hansen and C.-Y. Wang, *Contribution of third generation quarks to two-loop helicity amplitudes for W boson pair production in gluon fusion*, JHEP 01 (2021) 170 arXiv:2009.03742 [InSPIRE].

[65] P. Zhang, C. Meng, Y.-Q. Ma, and K.-T. Chao, *Gluon fragmentation into $3P_1^{[1,8]}$ quark pair and test of NRQCD factorization at two-loop level*, JHEP 08 (2021) 111 arXiv:2011.04905 [InSPIRE].

[66] C. Brønnum-Hansen and C.-Y. Wang, *Top quark contribution to two-loop helicity amplitudes for Z boson pair production in gluon fusion*, JHEP 05 (2021) 244 arXiv:2101.12095 [InSPIRE].

[67] R.-H. Wu, Y.-S. Zuo, C. Meng, Y.-Q. Ma, and K.-T. Chao, *NLO effects for $\Omega$ QQQ baryons in QCD Sum Rules*, Chin. Phys. C 45 (2021) 093103 arXiv:2104.07384 [InSPIRE].
[68] C. Brønnum-Hansen, K. Melnikov, J. Quarroz, and C.-Y. Wang, *On non-factorisable contributions to t-channel single-top production*, JHEP 11 (2021) 130 [arXiv:2108.09222] [InSPIRE].

[69] D. Baranowski, M. Delto, K. Melnikov, and C.-Y. Wang, *On phase-space integrals with Heaviside functions*, [arXiv:2111.13594] [InSPIRE].

[70] W.-L. Sang, F. Feng, Y. Jia, Z. Mo, and J.-Y. Zhang, *Next-to-next-to-leading-order QCD corrections to J/ψ + χc0,1,2 production at B factory*, [arXiv:2202.11615] [InSPIRE].

[71] E. Chaubey, M. Kaur, and A. Shivaji, *Master integrals for O(αs) corrections to H → ZZ*, [arXiv:2205.06339] [InSPIRE].

[72] Y.-D. Zhang, W.-L. Sang, and H.-F. Zhang, *Higher-order QCD corrections to Υ decay into double charmonia*, [arXiv:2205.06124] [InSPIRE].

[73] S. Abreu, M. Becchetti, C. Duhr, and M. A. Ozcelik, *Two-loop master integrals for pseudo-scalar quarkonium and leptonium production and decay*, [arXiv:2206.03848] [InSPIRE].

[74] F. Feng, Y. Jia, Z. Mo, J. Pan, W.-L. Sang, and J.-Y. Zhang, *Complete three-loop QCD corrections to leptonic width of vector quarkonium*, [arXiv:2207.14259] [InSPIRE].

[75] F. Feng, Y. Jia, Z. Mo, J. Pan, W.-L. Sang, and J.-Y. Zhang, *Three-loop QCD corrections to the decay constant of Bc*, [arXiv:2208.04302] [InSPIRE].

[76] W.-L. Sang, D.-S. Yang, and Y.-D. Zhang, *Z boson radiative decays to a P-wave quarkonium at NNLO and LL accuracy*, [arXiv:2208.10118] [InSPIRE].

[77] M. Beneke and V. A. Smirnov, *Asymptotic expansion of Feynman integrals near threshold*, Nucl.Phys. B522 (1998) 321–344 [hep-ph/9711391] [InSPIRE].

[78] V. A. Smirnov, *Problems of the strategy of regions*, Phys. Lett. B 465 (1999) 226–234 [hep-ph/9907471] [InSPIRE].
[79] A. I. Davydychev and J. B. Tausk, Two loop selfenergy diagrams with different masses and the momentum expansion, *Nucl. Phys. B397* (1993) 123–142 [InSPIRE].

[80] D. J. Broadhurst, Massive three - loop Feynman diagrams reducible to $SC^*$ primitives of algebras of the sixth root of unity, *Eur. Phys. J. C8* (1999) 311–333 [hep-th/9803091 [InSPIRE].

[81] Y. Schröder and A. Vuorinen, High-precision epsilon expansions of single-mass-scale four-loop vacuum bubbles, *JHEP 06* (2005) 051 [hep-ph/0503209] [InSPIRE].

[82] T. Luthe, Fully massive vacuum integrals at 5 loops. PhD thesis, Bielefeld U., 2015 [InSPIRE], https://pub.uni-bielefeld.de/publication/2776013.

[83] T. Luthe, A. Maier, P. Marquard, and Y. Schroder, Complete renormalization of QCD at five loops, *JHEP 03* (2017) 020 [arXiv:1701.07068] [InSPIRE].

[84] 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].

[85] M. Caffo, H. Czyz, M. Gunia, and E. Remiddi, BOKASUN: A Fast and precise numerical program to calculate the Master Integrals of the two-loop sunrise diagrams, *Comput. Phys. Commun. 180* (2009) 427–430 [arXiv:0807.1959] [InSPIRE].

[86] M. Czakon, Tops from Light Quarks: Full Mass Dependence at Two-Loops in QCD, *Phys. Lett. B664* (2008) 307–314 [arXiv:0803.1400] [InSPIRE].

[87] R. E. Cutkosky, Singularities and discontinuities of Feynman amplitudes, *J. Math. Phys. 1* (1960) 429–433 [InSPIRE].

[88] P. Maierhöfer and J. Usovitsch, Kira 1.2 Release Notes, [arXiv:1812.01491] [InSPIRE].