

Evaluation of Feynman integrals with arbitrary complex masses via series expansions.

Tommaso Armadillo^{a,b}, Roberto Bonciani^c, Simone Devoto^{b,d}, Narayan Rana^{b,d,e}, Alessandro Vicini^{b,d}

^a*Centre for Cosmology, Particle Physics and Phenomenology (CP3), Université Catholique de Louvain, Chemin du Cyclotron, B-1348 Louvain la Neuve, Belgium*

^b*Dipartimento di Fisica “Aldo Pontremoli”, University of Milano, Via Celoria 16, 20133 Milano, Italy*

^c*Dipartimento di Fisica, Università di Roma “La Sapienza” and INFN Sezione di Roma, Piazzale Aldo Moro 5, I-00185 Roma, Italy*

^d*INFN Sezione di Milano, Via Celoria 16, 20133 Milano, Italy*

^e*Department of Physics, Indian Institute of Technology Kanpur, 208016 Kanpur, India*

Abstract

We present an algorithm to evaluate multiloop Feynman integrals with an arbitrary number of internal massive lines, with the masses being in general complex valued, and its implementation in the **Mathematica** package **SeaFire**. The implementation solves by series expansions the system of differential equations satisfied by the Feynman integrals. At variance with respect to other existing codes, the analytical continuation of the solution is performed in the complex plane associated to each kinematical invariant. We present the results of the evaluation of the Master Integrals relevant for the NNLO QCD-EW corrections to the neutral-current Drell-Yan processes.

1. Introduction

The paper is organised as follows. In Section 2 we provide a pedagogical introduction to the solution of a system of first order linear differential equations by series expansions, with a particular focus on how the analytic continua-

Email addresses: tommaso.armadillo@uclouvain.be (Tommaso Armadillo), roberto.bonciani@roma1.infn.it (Roberto Bonciani), simone.devoto@unimi.it (Simone Devoto), narayan.rana@unimi.it (Narayan Rana), alessandro.vicini@mi.infn.it (Alessandro Vicini)

tion of the result can be safely obtained to an arbitrary point in the complex plane. We outline the implementation in the **Mathematica** package **SeaFire** of the algorithm that solves a generic system of differential equations, for arbitrary complex-valued kinematical variables and weights and we illustrate the different computational strategies available. In Section 3, as a first practical application of the algorithm, we discuss the solution of the Master Integrals needed to evaluate the two-loop QCD-EW virtual corrections to the neutral-current Drell-Yan processes. The latter constitute an original result of this paper. Finally, in Section 4 we draw our conclusions.

The latest version of the **Mathematica** package **SeaFire** can be downloaded from https://very_real_website.com, while its full documentation for Version 1.0 is provided in Appendix A.

2. The solution algorithm

The solution of a system of linear differential equations by series expansions is well known in the mathematical literature. In this Section we present some basic definitions and a pedagogical introduction to the procedure, and we show how it can be turned into an algorithm and applied to the specific case of Feynman integrals.

2.1. Solving differential equations by series expansion

The general method to obtain the solution as a series expansion can be illustrated with a simple example. Let us consider the differential equation

$$\begin{cases} f'(x) + \frac{1}{x^2 - 4x + 5} f(x) = \frac{1}{x+2} \\ f(0) = 1 \end{cases}; \quad (1)$$

with x being a real variable.

In order to solve the homogeneous equation we use the well-known *Frobenius method*. To this aim we introduce $f_{hom}(x) = x^r \sum_{k=0}^{\infty} c_k x^k$ as the expansion of the homogeneous solution around $x = 0$, with c_k being arbitrary coefficients that we need to determine. By replacing $f_{hom}(x)$ with its power series in the

associated homogeneous differential equation and by collecting all the terms with the same power of x , we obtain an infinite set of algebraic equations in the unknowns c_k . In our example, Eq.(1) leads to:

$$\left\{ \begin{array}{l} rc_0 = 0 \\ \frac{1}{5}c_0 + c_1(r+1) = 0 \\ \frac{4}{25}c_0 + \frac{1}{5}c_1 + c_2(2+r) = 0 \\ \frac{11}{125}c_0 + \frac{4}{25}c_1 + \frac{1}{5}c_2 + c_3(3+r) = 0 \\ \dots \end{array} \right. ; \quad (2)$$

The equation associated to the lowest power of x is called indicial equation, and it assigns the value of the exponent r . In our case, we get $rc_0 = 0$, which corresponds to $r = 0$. The other equations determine all the c_k but one; the latter can be chosen arbitrarily, and we set e.g. $c_0 = 5$. We thus obtain the solution to the homogeneous differential equation associated to the one in Eq.(1), expressed as the following series expansion:

$$f_{hom}(x) = 5 - x - \frac{3}{10}x^2 + \frac{11}{150}x^3 + \dots . \quad (3)$$

A particular solution for the original problem can now be obtained by applying the variation of the constant method, where the inverse of the homogeneous solution, multiplied by the inhomogeneous term, is expanded about $x = 0$ and easily integrated:

$$f_{part}(x) = f_{hom}(x) \int_0^x dx' \frac{1}{(x'+2)} f_{hom}^{-1}(x') = \frac{1}{2}x - \frac{7}{40}x^2 + \frac{2}{75}x^3 + \dots \quad (4)$$

The general solution is finally given by

$$f(x) = f_{part}(x) + Cf_{hom}(x) . \quad (5)$$

In order to satisfy the boundary condition $f(0) = 1$ we set $C = 1/5$.

2.2. Singularities and branch cuts

The validity of a solution obtained as a series expansion, with the procedure outlined in Section 2.1, is limited by its convergence radius. We now discuss

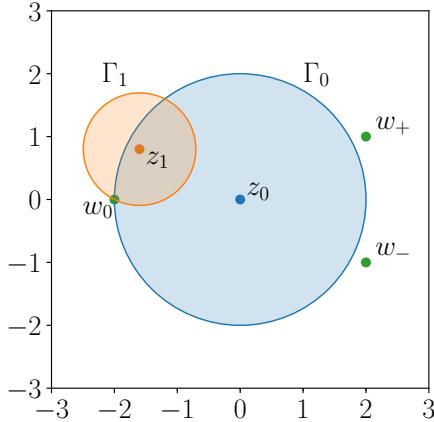


Figure 1: Example of analytic continuation.

how the latter is defined and how the solution can be extended, via analytic continuation, from its initial region of convergence to an external arbitrary point.

A solution written in series representation converges in the complex plane z inside a disc centered around the boundary condition point with a convergence radius limited by the closest singularity. As an example, we consider again the first-order linear differential equation presented in Eq.(1), now as a function of a complex variable z . By analysing the differential equation, we can identify three singularities: the factor $(z+2)$ in the denominator of the inhomogenous part generates a pole in $z = w_0 = -2$, while the factor $(z^2 - 4z + 5)$ in the denominator of the homogenous part implies the presence of two poles in $z = w_{\pm} = 2 \pm i$. The solution presented in Eq.(4) thus converges in the complex plane within a disc Γ_0 centered in $z_0 = 0$ with radius 2, as illustrated by the blue circle in Figure 1.

We can analytically continue the solution into a new disc, centered at an arbitrary point z_1 internal to Γ_0 . Also in this case, the convergence radius is defined by the closest singular point, as illustrated in Figure 1 by the orange circle Γ_1 . It is possible to demonstrate that this procedure is unique.

In the frequent physical case of simple poles in the homogeneous coefficient, we obtain a logarithmic behaviour of the solution, that requires some additional care in the analytic continuation. The presence of logarithmic functions makes

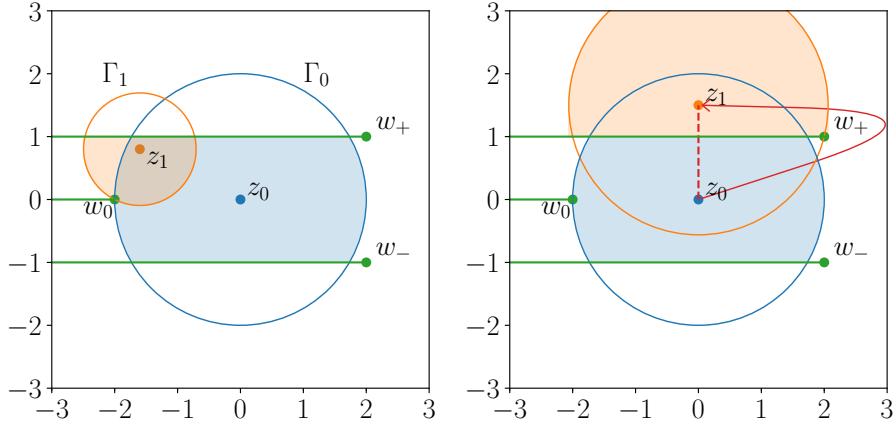


Figure 2: Example of the effect of branch-cuts on the convergence of the expanded solution: reduced convergence area (left) and different path for the analytic continuation (right).

the solution multi-valued: it can be made single-valued by adding cuts in the complex plane, thus specifying the Riemann sheet where the function is evaluated. This feature is clearly not present in the power series representation, but must be introduced to allow for a physical interpretation of the results. To take it into account, we need to define in a unambiguous way the cut associated to each singularity that shows a logarithmic behaviour: their presence will then affect the process of analytic expansion.

In our example, we now consider a logarithmic behaviour in the points w_0 , w_{\pm} , and we choose as the branch-cuts the horizontal lines parallel to the real axis that go from the singular point to $-\infty$. In the following discussion we will always assume the same convention, which represents the standard choice for logarithmic branch cuts, and the one already used in MATHEMATICA by the function `Log[z]`, for complex values of the variable z .

The first consequence of the presence of branch-cuts on the procedure of analytic continuation is on the area of convergence of the series expansion. While the proximity with a branch-cut does not modify the convergence radius of the series itself, it limits the area of the disc where the expansion converges to the desired value: once the cut is crossed, the series is evaluated on a Riemann

sheet different than the one that we have chosen with the cuts, and thus the result can not be considered as the solution of the problem in the assigned domain. This effect is illustrated on the left panel of Figure 2, that shows how the convergence discs Γ_0, Γ_1 introduced in Figure 1 are modified once the branch cuts are inserted: while the discs themselves do not shrink, only the reduced highlighted area provides the correct evaluation of the solution.

The second consequence is on the path that is necessary to follow in order to extend the solution to an arbitrary point in the complex plane. The path has to be chosen in such a way that it does not cross any branch cut. This is illustrated on the right panel of Figure 2, where the dotted path from $z = 0$ to z_1 is now forbidden by the presence of the branch cut corresponding to the pole w_+ , that needs to be avoided with a path that goes around to the right of the singularity, as shown by the filled line. For the sake of readability, we have not drawn all the discs along the path, representing the intermediate steps of the analytic continuation.

2.3. Choice of the path

In the MATHEMATICA package SEAFIRE, we use the same convention for the branch-cuts as the one we introduced in the previous section, with branch-cuts parallel to the real axis and going from the singularities to $-\infty$. As it has been shown in Section 2.2, with this choice the value of the solution in each point of the complex plane is unambiguously defined, as long as the path we use for the analytic continuation does not cross any branch-cut. In the following, we briefly describe how such path can be defined when connecting two arbitrary points on complex plane and the algorithm for its determination implemented within the package.

There are two main approaches that can be used while defining the path: we can avoid the singularities or we can use them as an expansion point. This two possibilities are illustrated in Figure 3, where we consider once again the pole structure given by Eq.(1), and we aim to move the boundary conditions from the point z_0 to the point z_1 .

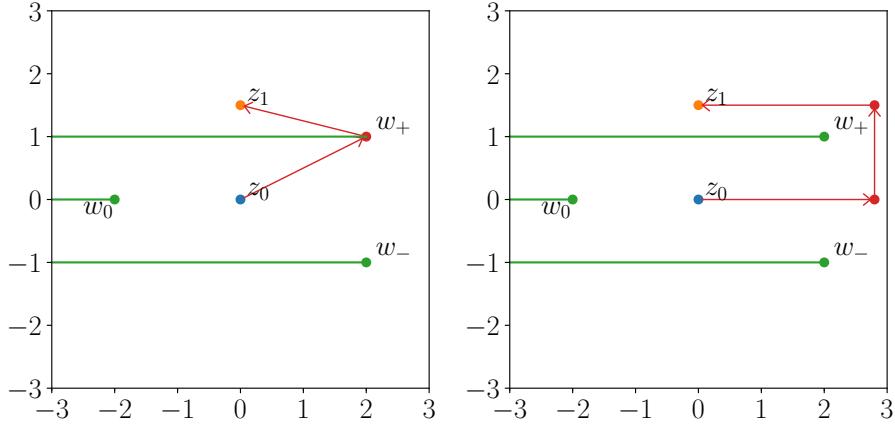


Figure 3: The two possible approaches in the definition of the path: the one requiring a logarithmic expansion (left) and the one only relying on the Taylor expansion (right).

By choosing the singularity in w_+ as one of the possible expansion points, as shown in the left panel of Figure 3, we obtain a logarithmic expansion around w_+ , while if we avoid the singularity, as shown in the right panel of Figure 3, we only rely on the ordinary Taylor expansion. While leading exactly to the same result, the two approaches have different advantages and disadvantages. By using the logarithmic expansion, we can expect a larger radius of convergence of the series, since the latter is not affected anymore by the presence of the pole we are using as our boundary condition: this can effectively reduce the number of steps required to reach the point of interest. Furthermore, the presence of the branch-cut starting from the singularity we are expanding upon is automatically taken into account by the explicit logarithms appearing in the series. On the other hand, in the current implementation of SEAFIRE, the expansion on a singular point requires a longer evaluation time with respect to the ordinary Taylor expansion on a regular point of the complex plane.

An optimised choice between the two different approaches thus requires to carefully balance this two effects. In the physical applications we dealt with while testing the code we observed how the presence of multiple singularities close to each other didn't allow for a significant improvement of the convergence

radius while using the logarithmic expansion, leading us to prefer, for our final implementation, the more regular behaviour of the ordinary Taylor expansion.

With this choice, the algorithm to determine the path is straightforwardly implemented. Since all the branch-cuts are parallel to each other and in the same direction, it is always possible to connect two arbitrary points on the complex plane with a path that goes around to the right of the rightmost singularity that lies between them, as it is shown in the right panel of Figure 3. While studying Feynman integrals with real internal masses, the poles are usually located on the real axis, that thus becomes a branch-cut starting from its rightmost singularity. Also the BCs and other points of interest usually assume real values, thus lying on the same branch-cut. The same situation might happen while considering the complex mass case: even if the poles are, in general, shifted on the complex plane, the boundary conditions and the point of interest might still accidentally lie on the same branch cut. If there is no pole between them, the path can simply connect them with a straight line: this approach does not lead to any ambiguity since the branch-cut is not problematic by itself, but it only leads to inconsistent results when it is crossed. On the other hand, an ambiguous situation is the one in which there is an additional pole located between them: if the path, in order to avoid the singularity, goes above or below the branch-cut, we obtain indeed two different results.

This ambiguity can be discussed and solved in the light of the Feynman prescription for the particle propagators, which enforces the causality requirement of the theory. If we consider, for the sake of simplicity, the propagator $\Pi(s) = i/(s - m^2 + i\varepsilon)$ of a massive scalar field, we see that the kinematic invariant s is real, but the Feynman prescription shifts it indeed to $s + i\varepsilon$ in the upper half of the complex s plane. We adopt the same prescription for the kinematic variables of the differential equation, which implies that the complete evolution of the solution from the BCs to the point of interest is performed not on the real axis, but in the complex plane of the kinematic variable, and, more precisely, in one specific half of that complex plane. Having a prescription which pushes the solution of the differential equation in one specific half of the z complex plane,

we implicitly choose the determination of the solution compatible with the BCs.

No Feynman prescription is hard-coded within SEAFIRE: as a consequence, while writing the differential equations, the user needs to provide as an input also the Feynman prescription for each variable they introduced, in order to resolve the ambiguity connected to such cases.

2.4. Solving a system of differential equations



The standard method to approach the evaluation of a Feynman integral is to construct a reduction algorithm that allows to write it as a linear combination of so called Master Integrals (MI). The computation of the MIs themselves is then usually faced by finding a differential equation, of which the MIs are a solution, and subsequently solve this equation.



For each MI, we can write the linear first-order differential equation with respect to one of the invariants which the integral depends upon. In the inhomogeneous term, we find additional integrals with simpler topologies, that can be written in terms of MIs by using the same reduction procedure previously introduced. The problem is thus reduced to the solution of the complete system of differential equations satisfied by the integral of interest and all the relevant subtopologies, collectively represented by $\vec{I}(\mathbf{s})$, where \mathbf{s} represents all the kinematic variables.

We have

$$\frac{\partial}{\partial s_\alpha} \vec{I}(\mathbf{s}) = \mathbf{A}_\alpha(\varepsilon, \mathbf{s}) \vec{I}(\mathbf{s}), \quad (6)$$

where s_α is a generic kinematic variable and the coefficient matrix \mathbf{A}_α contains rational functions of all the invariants.

The procedure outlined in section 2.1 to solve a single differential equation can be generalised in a straightforward way to a system of differential equations as the one in Eq.(6). Such an extension comes at the price of an increased complexity of the system for the coefficients equivalent to the one in Eq.(2), which will now contain the coefficients of the series expansion of all the unknown master integrals.

$\frac{\partial}{\partial S_\alpha}$

In general, the system of differential equations obtained in this way will depend on multiple kinematical variables, while we considered so far only one-dimensional differential equations. As a consequence, also the branch-cuts described in Section 2.2 actually become surfaces in a higher-dimensional space and, in order to describe the path for the transport of the boundary conditions, the expansion with respect to more variables can be in principle needed. In our implementation we circumvent these complications by considering the evolution of a single variable at a time. With this approach, during each single expansion we are dealing with a one-dimensional problem, and all the remarks made in the previous sections still hold.

An additional complication is given by the fact that, in practical applications, the computation is performed in dimensional regularisation, evaluating the integrals in $d = 4 - 2\epsilon$ dimensions. This introduce in the equations and in the solutions an additional dependence on the regularisation parameter ϵ which is in general non trivial and usually requires a series expansion in ϵ of the result in order to isolate its singular behaviour.

The system considerably simplifies, if it is possible, by an appropriate change of basis $\vec{I} \rightarrow \mathbf{B}\vec{I}$, to bring the coefficient matrix into the form $\mathbf{A} \rightarrow \epsilon \tilde{\mathbf{A}}$, called the canonical form. In this case we can write the system, in differential form, as

$$d\vec{I} = \epsilon d\tilde{\mathbf{A}} \vec{I}, \quad \tilde{\mathbf{A}} = \sum_l \tilde{\mathbf{A}}_l \log l$$

$\tilde{\mathbf{A}}$ does not depend on ϵ

with $\tilde{\mathbf{A}}_l$ a matrix of rational numbers and l the letters, i.e. the combination of kinematic variables which parametrise the singular structure of the scattering amplitude and in particular the various internal thresholds of the Feynman integral. The complete information about the singular structure of the problem under study, when it can be written in canonical form, is thus given by the set of the letters. The latter can be read from the matrix $\tilde{\mathbf{A}}$, in general after the application of a partial fractioning simplification procedure.

Writing the system in this form is particularly useful since automatically decouples the equations for each order in ϵ . We can thus start to solve from

ϵ the lowest order in the expansion and work our way up to the desired order. The MATHEMATICA package SEAFIRE, in its current implementation, can only solve system of differential equations provided in canonical form. Furthermore, at the moment it only accepts as an input a system provided in triangular form: thus the algebraic procedure to write the system as a triangular matrix, usually straightforward, needs to be performed by the user.

3. Two-loop MIs for the neutral-current Drell-Yan

In this section we present a first practical application of the algorithm previously discussed. In particular, we provide the results for the evaluation of the MIs relevant for calculating the mixed EW-QCD corrections for the neutral-current Drell-Yan, recently presented in Ref[]]. We firstly introduce the problem, then we show a comparison between the solution obtained with real and complex internal mass, and, finally, we discuss the numerical and time performances of our code.

3.1. Master integrals for the mixed EW-QCD corrections to the neutral-current Drell-Yan

The NNLO mixed EW-QCD corrections to the neutral current Drell-Yan have been calculated recently in Ref. [X]. In the complete amplitude 204 MIs appear and they can be evaluated using the method of differential equations. For a large subset of them, it is possible to solve their system of differential equations and express the solution in terms of generalised harmonic polylogarithms (GHPLs). For 5 of them, however, an analytical expression in terms of GHPLs does not exist and they can only be solved in terms of Chen-Goncharov integrals. These integrals belong to the following integral family: \mathcal{D}

$$\{\mathcal{D}_1, \mathcal{D}_2 - \mu_V^2, \mathcal{D}_{12}, \mathcal{D}_{1;1}, \mathcal{D}_{2;1}, \mathcal{D}_{1;12}, \mathcal{D}_{2;12} - \mu_V^2, \mathcal{D}_{1;3}, \mathcal{D}_{2;3}\} \quad (8)$$

where V can be either W or Z and μ_V^2 is the mass-squared of the V boson. We defined \mathcal{D} s to be:

$$\mathcal{D}_i = k_i^2, \mathcal{D}_{ij} = (k_i - k_j)^2, \mathcal{D}_{i;j} = (k_i - p_j)^2, \mathcal{D}_{i;jl} = (k_i - p_j - p_l)^2, \quad (9)$$

36 MIs
 with
 2 internal masses
~~X~~

where p_i are the external momenta and k_j the loop ones. In particular the 5 MI s are:

Keeping the numbering of ref C)

$$\begin{aligned} \text{Master 32} &: \{0, 1, 1, 1, 0, 1, 1, 0, 1\} & \text{Master 33} &: \{1, 1, 1, 1, 0, 1, 1, 0, 1\} \\ \text{Master 34} &: \{1, 1, 1, 1, -1, 1, 1, 0, 1\} & \text{Master 35} &: \{1, 1, 1, 1, 0, 1, 1, -1, 1\} \\ \text{Master 36} &: \{1, 1, 1, 1, -1, 1, 1, -1, 1\} \end{aligned} \quad (10)$$

It is possible to evaluate numerically the Chen iterated integrals in the euclidean region, where the boundary conditions are imposed. Their analytic continuation in the physical region, however, is non trivial and no standard technique to perform it is available. For this reason their computation in Ref.[] has been performed by using a semi-analytical approach, that is by solving the relevant system of differential equations using a series expansion method as described in Section 2. By approaching the numerical evaluation of these MI s in such a semi-analytical way, we have to solve the complete system (36x36) of differential equations which includes the integrals from eq. 10.

These MI s depend on two kinematic invariants, namely:

$$m \text{ or } \mu ? \quad s = \frac{(p_1 + p_2)^2}{m_V^2}, \quad t = \frac{(p_1 - p_3)^2}{m_V^2}, \quad (11)$$

where m_V is the mass of the vector boson, Z or W , inserted as a normalisation factor in order to handle dimensionless variables.

Once we have chosen the point (s, t) in which we would like to evaluate the solution, the latter can be obtained in a straight-forward way using **SeaFire**:

```
ConfigurationNCDY={  
    EpsilonOrder->4,  
    ExpansionOrder->50  
};  
UpdateConfiguration[ConfigurationNCDY];  
  
SetSystemOfDifferentialEquation[SystemOfEquations,  
    BoundaryConditions, MI $s$ , {s-I $\delta$ , t+I $\delta$ }, pointBC];
```

```

TransportBoundaryConditions[s, sValue]
TransportBoundaryConditions[t, tValue]

SolutionValue[]

```

In order to bring the system in canonical form, some master integrals have been scaled by a factor ϵ^k , with $k = 1, 2, 3, 4$. The final system goes from $\mathcal{O}(\epsilon^0)$ to $\mathcal{O}(\epsilon^4)$. The choice to expand up to $\mathcal{O}(\epsilon^4)$ is necessary in order to obtain the finite part for MI32-36 listed in eq. 10. We chose to keep 50 terms in the series because it is a good compromise between execution time and precision, we will comment later how the number of terms affects the numerical precision. In `SetSystemOfDifferentialEquation` we are preparing the package for solving the system and, finally, using `TransportBoundaryConditions` we are both solving the system and performing the analytic continuation, which let us extend the solution from `pointBC`, where the boundary conditions are imposed, to `s=sValue` and `t=tValue`. At this point, by calling again the routine `TransportBoundaryConditions` we could transport the boundary conditions from `{s=sValue, t=tValue}` to `{s=sValueNEW, t=tValueNEW}` and so on. By repeating this procedure multiple times we can easily obtain a numerical grid for all the master integrals. In Appendix A we provide the full documentation of the package.

3.2. Comparison of real and complex mass evaluations

When dealing with intermediate unstable particles, such as W s and Z s, it is useful to perform the calculations in the complex-mass scheme, in order to regularise the behaviour at **the resonance**. To this end, we introduce the complex mass, defined as:

$$\mu_V^2 = m_V^2 - i\Gamma_V m_V, \quad (12)$$

where Γ_V , a real parameter, labels the decay rate of the boson. The complex mass μ_V then replaces the real mass m_V in all the steps of the computation.

From Eq. (11), in particular, we can observe that the dimensionless kinematic variables s and t get complex-valued.

Its net effect is a regularisation of the integrand function in the threshold region: indeed, all the propagators

$$\frac{1}{s - \mu^2 + i\delta} \quad (13)$$

real
do not contain anymore any divergences for any value of s , and, hence, in the kinematic region near the resonances, the MIs are smooth functions of the kinematic variables.

This fact let us really appreciate the discussion about the analytic continuation in the complex plane presented in Section 2: thanks to the original algorithm for performing the analytic continuation of the solution included in SEAFIRE, we are able to evaluate the 5 MIs of interest in every point of the physical region and for arbitrary complex variables, thus implementing the complex-mass scheme in a straightforward way.

In Fig. 4 we report the comparison between real- and complex-valued internal masses for six selected MIs, evaluated with SEAFIRE. The plots present the MIs as a function of \sqrt{s} at fixed $\cos \theta = 0$, where we label with θ the scattering angle in the centre of mass reference frame. The latter can be written in terms of the kinematic invariant t introduced in Eq. (11) with the following relation:

$$t = \frac{-s}{2} (1 - \cos \theta). \quad (14)$$

For each plot in Fig. 4, the blue lines represent the masters for real-valued masses, while the yellow the complex-valued ones. The solid and dashed lines represent the real and imaginary part of the solution, respectively.

The first plot shows the result for MI 10, defined as follows:

$$\text{Master 10 : } \{\dots\}. \quad (15)$$

It provides a validation of our code: for the first 31 MIs, indeed, we have an analytic expression in terms of GHPLs that can be checked against SEAFIRE, also for complex values of the internal mass. In the plot we can observe an

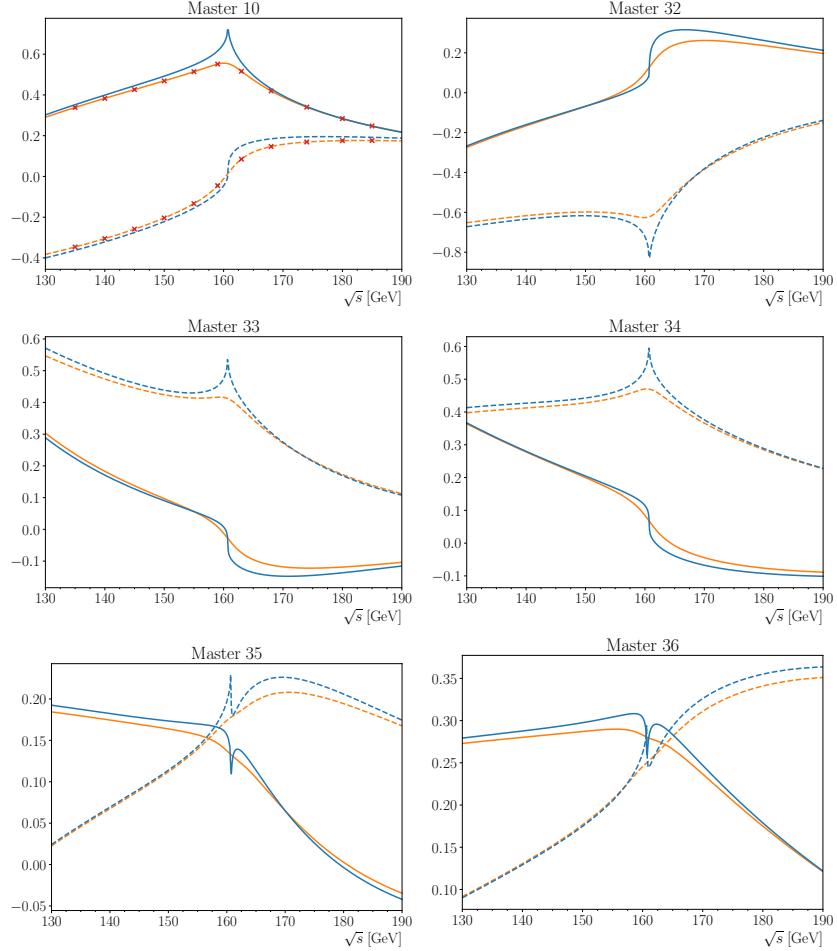


Figure 4: Comparison between real and complex valued-masses for MI10 and MIs from 32 to 36. The plots show the value of the finite ϵ -order for $130 \text{ GeV} \leq \sqrt{s} \leq 190 \text{ GeV}$, $\cos \theta = 0$ and $\mu^2 = m_W^2$ or $\mu^2 = m_W^2 - im_W \Gamma_W$. The blue lines represent the masters for real-valued masses, while the yellow the complex-valued ones. The solid and dashed lines represent the real and imaginary part of the solution, respectively. In the MI10 plot, the red crosses represent the analytical value obtained with **GiNaC**.

excellent agreement between our result and the red crosses, that represent the value of the analytic result, evaluated with GINAC for a few selected points.

The remaining 5 plots show the comparison between real- and complex-valued internal masses for MI32-36, and represent an original result of this

paper. We can observe that the masters with real-valued masses exhibit a non differentiable behaviour around $\sqrt{s} = 2m_W$, while they become smooth when moving to the complex-mass scheme, making the latter particularly important for phenomenological studies in the threshold region.

3.3. Performances

As anticipated in the previous section, we can check our numerical results against other publicly available tools, namely **GiNac** and **DiffExp**. In particular, we can use **DiffExp** to check the value of the 36 MIs considering internal real-valued masses. We can use **GiNaC** to check the first 31 masters with internal complex-valued masses. The value of the last 5 MIs with internal complex masses is, instead, a new prediction.

From a practical point of view evaluated the masters in 6 different points of the phase-space, i.e. $\sqrt{s} \in \{70 \text{ GeV}, 155 \text{ GeV}, 220 \text{ GeV}\}$, $\cos \theta \in \{0, 0.75\}$ and $\mu^2 \in \{m_W^2, m_W^2 - im_W\Gamma_W\}$. The points were chosen such that we have a sample from every region between the resonances, i.e. $\sqrt{s} < m_W$, $m_W < \sqrt{s} < 2m_W$ and $\sqrt{s} > 2m_W$. The numerical results are shown in Table 1.

Phase-space point	Comp. DiffExp (MI1-36, real mass)	Comp. GiNaC (MI1-31, complex mass)
$\sqrt{s} = 70 \text{ GeV}, \cos \theta = 0$	10^{-19}	10^{-15}
$\sqrt{s} = 70 \text{ GeV}, \cos \theta = 0.75$	10^{-20}	10^{-15}
$\sqrt{s} = 155 \text{ GeV}, \cos \theta = 0$	10^{-21}	10^{-16}
$\sqrt{s} = 155 \text{ GeV}, \cos \theta = 0.75$	10^{-21}	10^{-16}
$\sqrt{s} = 220 \text{ GeV}, \cos \theta = 0$	10^{-21}	10^{-16}
$\sqrt{s} = 220 \text{ GeV}, \cos \theta = 0.75$	10^{-21}	10^{-16}

Table 1: These numbers represent the absolute difference between our results and other existing packages. In particular we are solving the 36 MIs from $\mathcal{O}(\epsilon^0)$ up to $\mathcal{O}(\epsilon^4)$, then, for everyone of the 180 coefficients we obtain, we take the difference with the expected value and we evaluate its absolute value. The number reported above is the maximum among these differences. **DiffExp** and **SeaFire** results are obtained considering 75 terms in the series expansion.

Regarding the time performances of **SeaFire**, they strongly depend on how close are the starting and ending point, and of course on the dimension of the system. In the case of the mixed EW-QCD correction for the neutral current Drell-Yan every step, that is solving the system and evaluating the solution in the following point, requires ~ 50 second. This estimate is obtained on a MacBook Pro (2015) considering 50 terms in the series expansion. For transporting the boundary conditions from the euclidean region to the physical one, usually 15 – 20 steps are necessary. Once in the physical region, less singularities are present and, hence, more direct path can be chosen. In this case a number between 1 and 10 steps is sufficient to reach every point in the phase-space.

The numerical grid constituted by 3250 points and presented in [?] have been evaluated on an Intel Xeon Silver 4110. The calculation was performed on 26 cores and took 9 hours in total. We stress that, in our case, the evaluation of the numerical grid for the MIs is a process that has to be done only once. In fact we could use the above grid for evaluating a grid for the total correction, and then, thank to its smoothness we can obtain the corrections in any point of the phase space by a linear interpolation. The estimate on the error for the latter is 10^{-3} s.

4. Conclusions

Appendix A. Package documentation

The **SeaFire** package (SEmi-Analytical Feynman Integrals RESolution) can be imported in Mathematica using the command `Get[...]`, i.e. through `<< SeaFire.m`. Note that it has been developed and tested on Mathematica 12.0.

Below we present all different functions and their functionalities.

- `CurrentConfiguration[]`

It returns the current configuration of the package. The configuration parameters can be modified using the function `UpdateConfiguration`.

- `UpdateConfiguration[NewConfig]`

`NewConfig` must be a `List` whose elements are replacement rules `NameParameter` \rightarrow `NewValue`. See Table A.2 for a complete overview on all the parameters that the user can modify.

- `ReadFrom[FilePath_]`

Utility function which reads from the specified path and return the content of the file. X

- `SetSystemOfDifferentialEquation[System_, BCs_, MIs_, Variables_, PointBC_, Param_:{}]`

It sets all the internal variables of the package and prepare the system of differential equations. It receives in input

– `System_`: the system of differential equations. The equations must be ordered so that the first one is the simplest one. Moreover, the system must be in canonical form. If there are multiple kinematics variables, one must pass firstly the system for the first variable, then for the second one and so on, where the order is specified in the `Variables_` parameter. The equations can be given expanded in ϵ or in a closed form in ϵ . ? X ?

Example:

```
{ B1(1,0)[s,t] == 0 ,
  B2(1,0)[s,t] == (- 1/s - ε/s)B2[s,t] - ε B1[s,t] / s,
  B1(0,1)[s,t] == 0 ,
  B2(0,1)[s,t] == 0 }
```

– `BCs_`: the boundary conditions for the given equation. They can be given in a closed form in ϵ or as a series. They can also be given as an asymptotic limit. They can be exact or floating numbers, in this case make sure that the precision of the boundary condition is sufficient for your final precision goal.

Example:

```
{ B1[1,1] == - 1/32 + ε/32 + ε²( - 1/32 + π²/96 ),  
B2[1,1] == - ε Log[2]/16 + ε²(-π²/192 + Log[2]²/8) }
```

- **MIs_**: the list of master integrals, as they appear in the equations.

Example:

```
{ B1[s,t], B2[s,t] }
```

- **Variables_**: the list of variables appear in the equations, together with their Feynman prescriptions $\pm i\delta$

Example:

```
{ s + I δ, t + I δ }
```

- **PointBC_**: the point in the phase-space in which the boundary conditions are imposed.

Example:

```
{ 1, 1 }
```

- **Param_**: this is an optional parameters. Some equations might contain some external parameters, for example some masses **Mw**, **Mz**. This substitutions are performed before solving the system.

Example:

```
{ MW -> 80.38, MZ -> 91.19 }
```

- **GetSystemOfDifferentialEquation[]** and
GetSystemOfDifferentialEquationExpanded[]

They return the system of differential equations before and after it has been expanded in ϵ . These functions can be used to check if everything has been set correctly.

- **SolveSystem[Variable_]**

It solves the system of differential equations in the current point, that is where the boundary conditions are currently imposed.

- **GetPoint[]**

It returns the current point in which the boundary conditions are imposed.



- `TransportBoundaryConditions[Variable_, Destination_, Line_:{}]`

It transports the boundary conditions for the variable `Variable` from the current point to `Destination`. After transporting the boundary conditions to a certain point, the point in which the boundary conditions are imposed is updated to `Destination`. The `Line` parameter is optional. If the user is not satisfied by the path automatically chosen by the package can use his own. The `Line` object must be created with `CreateLine`

their

- `CreateLine[Points_]`

It returns a line object that can be used in `TransportBoundaryConditions`

- `Solution[]`

It returns the series solution in the current point. The coefficients of the series are given with `InternalWorkingPrecision` digits. The result is given as a `List` and every MI as a Laurent series in ϵ .

- `CheckSingularities[]`

It checks whether the singularities are logarithmic, i.e. if they develop a branch-cut. The check is done by performing a path round the singularity and checking if the solution is the same. If it is not, it means that in doing so we crossed a branch-cut and, hence, the singularity is logarithmic. If this function is not called, `SeaFire` consider every singularity as logarithmic, and the analytic continuation is still possible. However, if the user is planning to make an intensive use of the function `TransportBoundaryConditions`, e.g. create a numerical grid, knowing the position of not-logarithmic singularities might allow for more direct paths in the complex plane. The output of `CheckSingularities` can be passed back in the `SafeSingularities` and `LogarithmicSingularities` parameters for future runs.

✓

✗

- `CreateGraph[MI_, EpsOrder_, Left_, Right_, OtherFunctions_:{}]`

~~~~~

for + ?

Draws a `ReImPlot` of the solution of order `EpsOrder` for the master `MI`.
The graph runs from `Left` to `Right`. The argument `OtherFunctions` may
contains other functions to be plotted in the same graph.

In the `Example/` folder of the GitHub repository of `SeaFire`, the user can
find working examples to play with.

NameParameter	Value type	Default	Description
EpsilonOrder	Integer	2	The maximum order in the dimensional regulator ϵ at which the system is expanded. Note that the minimum order is determined by the boundary conditions, i.e. if they contain a term $1/\epsilon^2$ the minimum order will be -2 .
ExpansionOrder	Integrer	50	The maximum order in the kinematic variable at which the solution is expanded.
InternalWorkingPrecision	Integer	250	Specifies the number of digits that are used in internal calculations. If it is too high, the execution will require more time and space, if it is too low, we may face rounding errors.
LogarithmicExpansion	Bool	False	Specifies which method to use for transporting boundary conditions. If it is set to True, SeaFire will expand also on top of singularities.
RadiusOfConvergence	Integer	2	It controls how fast we move at every expansion. If RadiusOfConvergence is n , and the maximum radius of convergence of the solution is r , then the new point will be distant r/n from the center of the series. Note that r is determined internally at every step, based on the position of singularities.
LogarithmicSingularities	List	{}	The user can explicitly state which singularities are of Logarithmic type, i.e. develop a branch-cut, and which not. This might speed up the evaluation of numerical grids since it allows more direct paths in the phase-space. The format in which the singular points are passed must be the one returned by CheckSingularities[] . X
SafeSingularities	List	{}	Same as LogarithmicSingularities .

Table A.2: All the parameters that can be modified by the user.