RISC-Linz
Research Institute for Symbolic Computation
Johannes Kepler University
A-4040 Linz, Austria, Europe

First-order factorizable systems of differential equations in one variable

N. Fadeev

November 2020

RISC-Linz Report Series No. 20-20

ESR Progress Report

First-order factorizable systems of differential equations in one variable

Nikolai Fadeev[I]

Under the supervision of Carsten Schneider[II]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764850 (SAGEX)

Research Insitute for Symbolic Computation (RISC)
RISC Software GmbH
November 26, 2020

[I]E-mail: nfadeev@risc.jku.at
[II]E-mail: carsten.schneider@risc.jku.at

Contents

1 Introduction

Developped in the first half of the XX$^{\text{th}}$ century, quantum field theory (QFT) is one of the most powerful modern theories that describes the world of the very small. Designed originally to deal with relativistic quantum mechanics, it came out to be a general framwork for dealing with many-particle problems that could also provide a consistent quantum mechanical description for most of the fundamental forces – electromagnetism, the strong force and the weak one. At the heart of the theory lies the notion of scattering amplitude, i.e. the probability amplitude for scattering from some initial state $|i\rangle$ with m particles to some final state $|f\rangle$ with n particles. In interacting theories, scattering amplitudes are often defined perturbatively and computing them to higher order is necessary to better understand the theory and make more precise experimental predictions.

For involved QFTs such as Yang-Mills theories, these computations can be particularly difficult when we are going to higher numbers of loops, and are usually carried in the following way: taking the traditional approach, one can write all the Feynman diagrams with their particular color, momentum and Dirac structure and then reduce them all to some master integrals [24, 25, 27, 32, 33, 35–37]. These will depend typically on two parameters: some x that might be a Feynman parameter or a Landau variable on which depends the center-of-mass energy s, but also the ϵ variable, coming from the dimensional regularisation (or any other regularisation scheme).

It so happens that the method used to reduce the many Feynman diagrams with given topology to master integrals, called "integration by parts" (IBP) method, provides also another way to compute them by generating a system of first order differential equations in the x variable for the said master integrals [3, 28, 30, 39]. The question is therefore simple: how to solve this first order differential system order by order in the ϵ parameter?

In this report, we present two methods that allow to do so. After some preprocessing of the system that allows to reduce it to simpler subproblems (section 2), the first method consists in expanding the system in ϵ and solving the subsequent series of systems by uncoupling them [10] (section 3), wheareas the second one uncouples them first and then proceeds to expand the obtained higher order differential equation in ϵ to solve it iteratively [45] (section 4), using in particular difference ring and field machinery. Both methods are mostly designed for systems obeying some constraints (in particular they must be first-order factorizable). Expanding the scope of those as well as implementing extensions for new classes of functions will be part of future work that will be explicitely outlined (section 5).

2 Preprocessing the system

From now on, we denote the master integrals of our system as $\mathcal{I}_i(x,\epsilon)$, and we define the vector containing all of them as $\mathcal{I}(x,\epsilon) := (\mathcal{I}_1(x,\epsilon),\dots,\mathcal{I}_n(x,\epsilon)^\top$, where n is the total number of master integrals we are dealing with as well as the size of our system obtained from IBP. The latter is defined therefore as follows:

$$\frac{\mathrm{d}}{\mathrm{d}x}\mathcal{I}(x,\epsilon) = \mathcal{M}(x,\epsilon)\mathcal{I}(x,\epsilon) + \mathcal{R}(x,\epsilon) \tag{1}$$

where $\mathcal{M}(x,\epsilon)$ is the matrix of coefficients and $\mathcal{R}(x,\epsilon) := (\mathcal{R}_1(x,\epsilon),\dots,\mathcal{R}_n(x,\epsilon))^\top$ is the inhomogeneous part of the system. Due to the IBP reduction principles, $\mathcal{M}(\epsilon,x) \in \mathcal{M}_n(\mathbb{K}(x,\epsilon))$, i.e. the coefficient matrix is defined over the field of rational functions in ϵ and x over $\mathbb{K}$, where $\mathbb{K}$ is a computable subfield of $\mathbb{R}$ (for instance $\mathbb{Q}$). As for the inhomogeneous part, it might contain new Feynman integrals obtained when integration by parts produces integrals that are not part of our $\mathcal{I}(x,\epsilon)$ master integrals – those will be called base integrals (BI) in the following, and we suppose that some expansion of those is already known, or can be obtained[1].

2.1 Solving hypotheses

In order to solve our system, it has to verify several general assumptions, which are the following:

- The coefficient matrix $\mathcal{M}(x,\epsilon)$ has no poles around $\epsilon = 0$, i.e. it has a Taylor expansion of the following form:

$$\mathcal{M}(x,\epsilon) = \sum_{i=0}^{\infty} \mathcal{M}^{(i)}(x)\epsilon^i. \tag{2}$$

  This form of expansion can actually always be achieved by index-shift. More precisely, suppose that $\mathcal{M}(x,\epsilon)$ has a pole of order k in ϵ, i.e.

$$\mathcal{M}(x,\epsilon) = \sum_{i=-k}^{\infty} \mathcal{M}^{(i)}(x)\epsilon^i = \frac{\mathcal{M}^{(-k)}(x)}{\epsilon^k} + \cdots + \sum_{i=0}^{\infty} \mathcal{M}^{(i)}(x)\epsilon^i.$$

[1]For instance, one could apply recursively the IBP method and/or use symbolic summation/integration algorithms (see [20] and the literature therein) in order to obtain new systems or closed form representations.

Then, since $\mathcal{M}(x,\epsilon)$ only appears in the equation (1) as the matrix multiplying the MI vector $\mathcal{I}(x,\epsilon)$, we can do the following redefinition of the latter:

$$\sum_{i=-k}^{\infty}\mathcal{M}^{(i)}(x)\epsilon^i\mathcal{I}(x,\epsilon)=\sum_{i=-k}^{\infty}\mathcal{M}^{(i)}(x)\epsilon^{i+k}\left(\frac{\mathcal{I}(x,\epsilon)}{\epsilon^k}\right)=\underbrace{\tilde{M}(x,\epsilon)}_{=:\sum_{i=0}^{\infty}\mathcal{M}^{(i-k)}(x)\epsilon^i}\underbrace{\tilde{\mathcal{I}}(x,\epsilon)}_{=:\frac{\mathcal{I}(x,\epsilon)}{\epsilon^k}}.$$

This means that poles of $\mathcal{I}(x,\epsilon)$ will be shifted, but this is not a problem because of the second assumption.
In addition, after such a shift has been achieved, we suppose that

$$\det\mathcal{M}^{(0)}(x)\neq 0.$$

- Each element of $\mathcal{R}(x,\epsilon)$ can be Laurent-expanded in ϵ and therefore the solutions $\mathcal{I}(x,\epsilon)$ that we search are also assumed to be expandable in the same way, i.e.

  $$\exists k\in\mathbb{N}\quad\forall i\in\{1,\dots,n\}\quad\mathcal{R}_i(x,\epsilon)=\sum_{j=-k}^{\infty}\mathcal{R}_i^{(j)}(x)\epsilon^j,\quad\mathcal{I}_i(x,\epsilon)=\sum_{j=-k}^{\infty}\mathcal{I}_i^{(j)}(x)\epsilon^j.$$

  Therefore, the previous point makes sense – if $\mathcal{M}(x,\epsilon)$ has poles around $\epsilon=0$, we index-shift it so that the poles of $\mathcal{I}_i(x,\epsilon)$ are themselves shifted and this is fine, since we suppose they have poles anyway.

- The given coefficients $\mathcal{R}_i^{(j)}(x)$ are polynomial expressions in terms of hyperexponential functions and iterated integrals over hyperexponential functions, and therefore we seek for solution coefficients $\mathcal{I}_i^{(j)}(x)$ in the same function space.

Let us now define what is exactly an hyperexponential function.

Definition 2.1 (Hyperexponential function). *Let $\mathbb{K}$ be some (computable) field. A non-zero function $f(x)$ over $\mathbb{K}$ is called **hyperexponential** if there exists some non zero $r(x)\in\mathbb{K}(x)$ such that*[2]

$$\frac{\mathrm{d}_x f(x)}{f(x)}=r(x).$$

We will denote the set of hyperexponential functions over $\mathbb{K}$ with $\mathcal{H}^e(\mathbb{K})$.

Examples: The class of hyperexponential functions is relatively large, it includes for instance:

[2]We note as an abreviation $\frac{\mathrm{d}}{\mathrm{d}x}\equiv\mathrm{d}_x$.

- all rational functions over $\mathbb{K}$, i.e. $\mathbb{K}(x) \subset \mathcal{H}^e(\mathbb{K})$;
- more generally, all rational functions to some power, i.e.

$$\{f^q(x) \mid f(x) \in \mathbb{K}(x),\ q \in \mathbb{K}\} \subset \mathcal{H}^e(\mathbb{K}).$$

  In our case and for our computations, this reduces usually to rational functions to some rational power, i.e. $q \in \mathbb{Q}$;
- the exponential function exp itself.

The last point prompts us to write another possible definition of an hyperexponential function $f(x)$ as an exponentiated integral over a rational function $r(x)$:

$$\exists l \in \mathbb{K},\ f(x) = e^{\int_l^x \mathrm{d}y\, r(y)}.$$

From the hyperexponential function one might extend naturally to iterated integrals over such functions:

Definition 2.2 (Iterated integral over hyperexponential functions)**.** *Let $\mathbb{K}$ be a (computable) field, $l_0, \ldots, l_n \in \mathbb{K}$ and $f_1, \ldots, f_n \in \mathcal{H}^e(\mathbb{K})$. Then the following integral*

$$I(x) = \int_{l_0}^{x} \mathrm{d}x_1 f_1(x_1) \int_{l_1}^{x_1} \mathrm{d}x_2 f_2(x_2) \cdots \int_{l_{n-1}}^{x_{n-1}} \mathrm{d}x_n f_n(x_n) \text{ with } l_0, \ldots, l_n \in \mathbb{K}$$

is called an ***iterated integral over hyperexponential functions***[3]*. We will denote the set of IOH over $\mathbb{K}$ with $\mathcal{IH}^e(\mathbb{K})$.*

The set $\mathcal{H}^e(\mathbb{K})$ is also especially nice since it is in particular stable under:

- Multiplication: take $f(x), g(x) \in \mathcal{H}^e(\mathbb{K})$ such that there exists $r(x), s(x) \in \mathbb{K}(x)$ such as:

$$\frac{\mathrm{d}_x f}{f} = r,\ \frac{\mathrm{d}_x g}{g} = s.$$

  Then

$$\frac{\mathrm{d}_x(fg)}{fg} = \frac{(\mathrm{d}_x f)g}{fg} + \frac{f(\mathrm{d}_x g)}{fg} = \frac{\mathrm{d}_x f}{f} + \frac{\mathrm{d}_x g}{g} = r + s \in \mathbb{K}(x). \quad \square$$

- Inversion: take $f(x) \in \mathcal{H}^e(\mathbb{K})$ with $r(x) \in \mathbb{K}(x)$ such that $\frac{\mathrm{d}_x f}{f} = r$. Then

$$\frac{\mathrm{d}_x\left(\frac{1}{f}\right)}{\left(\frac{1}{f}\right)} = f \times \left(-\frac{\mathrm{d}_x f}{f^2}\right) = -\frac{\mathrm{d}_x f}{f} = -r \in \mathbb{K}(x). \quad \square$$

[3]Abreviated as IOH from now on.

- Differentiation: take $f(x) \in \mathcal{H}^e(\mathbb{K})$ with $r(x) \in \mathbb{K}(x)$ such that $\frac{\mathrm{d}_x f}{f} = r$. Then

$$\frac{\mathrm{d}_x^2 f}{\mathrm{d}_x f} = \frac{\mathrm{d}_x(rf)}{rf} = \frac{(\mathrm{d}_x r)f}{rf} + \frac{r(\mathrm{d}_x f)}{rf} = \frac{\mathrm{d}_x r}{r} + r \in \mathbb{K}(x). \quad \square$$

Moreover, due to the structure of the shuffle algebra of iterated integrals over hyperexponential functions[4], any polynomial expression out of hyperexponential functions and IOH can always be written as a linear combination of the form

$$f_1(x)I_1(x) + \cdots + f_n(x)I_n(x) \text{ with } h_i(x) \in \mathcal{H}^e(\mathbb{K}),\ I_i(x) \in \mathcal{IH}^e(\mathbb{K}).$$

Note: The functions that are not included in this set are for instance some classes of special functions, such as the elliptic functions – those must be dealt with differently, and we plan to investigate this part in the future.

2.2 Triangularization

When applying the IBP method [3, 28, 30, 39], we usually end up with a system with a rather high size n, usually of order $n \sim 100$. Nevertheless, from a physical point of view, there is an argument that allows to reorganize it in a way that is advantageous for us. Indeed, the MI are classified in what is called "sectors", i.e. sets of maximum number of non-vanishing propagators in a single Feynman graph. When some propagator is absent, this defines a sub-sector, and a differential equation for a MI only contains integrals from the same sector or its sub-sectors. What this technically entails at our level is that one can find several sub-systems where the MIs depend only on themselves and on MIs from lower sub-systems (so that they can be considered effectively as BIs in this sub-system). More precisely, one can perform the following pre-processing step:

Step 0: We "triangularize" the system in n smaller sub-systems where the ith subsystem of coupled MIs depend only on the said MIs and on those from the systems 1 through $i - 1$.

Note: This step is *per se* not included in the algorithm and relies once more on the IBP machinery, but is is of uttermost importance as it usually reduces a unique problem of size $n \sim 100$ to several recursive problems of size $n \lesssim 10$ that are much easier to deal with.

[4]See for instance [14].

3 Solving iteratively the system

3.1 Defining recursive sub-systems

We will denote from now on the subsystems with a tilde, i.e. $\tilde{\mathcal{I}} = (\tilde{I}_1(x), ..., \tilde{I}_n(x))$ ($n \lesssim 10$), and we study now a subsystem MI obtained after such a step:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon). \tag{3}$$

Given the expansion assumptions of the previous section, we can Laurent-expand the MI, BI and the coefficient matrix[5]:

$$\tilde{\mathcal{M}} = \sum_{j=0}^{\infty} \tilde{M}^{(j)}\epsilon^j, \quad \tilde{\mathcal{R}} = \sum_{j=-k}^{\infty} \mathcal{R}_i^{(j)}\epsilon^j, \quad \tilde{\mathcal{I}} = \sum_{j=-k}^{\infty} \mathcal{I}_i^{(j)}\epsilon^j. \tag{4}$$

Step 1: Plugging (4) in (3), we can collect the terms with the same powers of ϵ, which gets us an infinite-dimensional series of differential equation systems (DES):

$$\begin{cases} \mathrm{d}_x\tilde{\mathcal{I}}^{(-l)} & = & \tilde{\mathcal{M}}^{(0)}\tilde{\mathcal{I}}^{(-l)} + \tilde{\mathcal{R}}^{(-l)} & \epsilon^{-l} \\ \mathrm{d}_x\tilde{\mathcal{I}}^{(-l+1)} & = & \tilde{\mathcal{M}}^{(0)}\tilde{\mathcal{I}}^{(-l+1)} + (\tilde{\mathcal{M}}^{(1)}\tilde{\mathcal{I}}^{(-l)}) + \tilde{\mathcal{R}}^{(-l+1)} & \epsilon^{-l+1} \\ \vdots & & \vdots & \vdots \\ \mathrm{d}_x\tilde{\mathcal{I}}^{(k)} & = & \tilde{\mathcal{M}}^{(0)}\tilde{\mathcal{I}}^{(k)} + \left(\sum_{i=1}^{k+l}\tilde{\mathcal{M}}^{(i)}\tilde{\mathcal{I}}^{(k-i)}\right) + \tilde{\mathcal{R}}^{(k)} & \epsilon^{k} \\ \vdots & & \vdots & \vdots \end{cases} \tag{5}$$

By doing this step, we've eliminated the ϵ dependence from the system, so what remains is a set of recursively dependent systems of first order that depend on x. Let us see how we can iteratively solve these in a bottom-up approach. We will take for the example the first system of the set, i.e.

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}^{(-l)} = \tilde{\mathcal{M}}^{(0)}\tilde{\mathcal{I}}^{(-l)} + \tilde{\mathcal{R}}^{(-l)}. \tag{6}$$

Remark: Even though it seems the easiest choice, it is actually perfectly representative of the general method. Indeed, we see that in all the subsystems, the coefficient matrix is always $\mathcal{M}^{(0)}$, and only the inhomogeneous part changes, including each

[5]From now on, since we suppressed the dependence on x of the coefficients, we will omit the (x).

time new contributions from the previous MI coefficients $\tilde{\mathcal{I}}^{(k)}$. Therefore, solving this once will provide us with the general steps and solutions to solve each higher DES.

3.2 Uncoupling the system

In order to solve the system, the first (and natural) step is to uncouple it, i.e. to transform a first-order system of n equations in n MIs into a single inhomogeneous higher-order differential equation (HODE) in one of the MI, say $\tilde{\mathcal{I}}_1^{(-l)}$, where the inhomogeneous part depends on the inhomogeneous part of the system, combined with $n-1$ equations relating linearly over the field of rational functions the other MIs to the first one, its derivatives and the inhomogeneous part of the system. In all those linear relations, the coefficients are from the field $\mathbb{K}(x)$. In order to do the uncoupling step, we use the packages `OreSys` [26], written by Stefan Gerhold and optimised by Carsten Schneider for our problem.

Step 2: Using the `OreSys` package, we uncouple the sub-system:

$$(6) \overset{\text{OreSys}}{\Longrightarrow} \left\{ \begin{array}{c} \boxed{\sum_{k=0}^{m} p_k(x) \mathrm{d}_x^k \tilde{\mathcal{I}}_1^{(-l)}(x) = r(x)} \\ \Lsh \text{ HODE in } \tilde{\mathcal{I}}_1^{(-l)}(x),\ p_i(x) \in \mathbb{K}(x) \\ r(x) = \sum_{i=0}^{p} \sum_{j=1}^{m} r_{i,j}(x) \mathrm{d}_x^i \tilde{\mathcal{R}}_j^{(-l)}(x) \\ \Lsh\ p \in \mathbb{N},\ r_{i,j}(x) \in \mathbb{K}(x) \\ \tilde{\mathcal{I}}_j^{(-l)}(x) = \sum_{i=0}^{m-1} a_{k,i}(x) \mathrm{d}_x^i \tilde{\mathcal{I}}_1^{(-l)}(x) + \rho_k(x) \\ \Lsh\ j \in \{2, \ldots, m\},\ a_{k,i}(x) \in \mathbb{K}(x),\ \rho_k(x) \text{ like } r(x) \end{array} \right. \quad (7)$$

Note: There exists several possible uncoupling algorithms in the package `OreSys` – Gauß, Euclid, Zürcher [16, 22, 46] among others – and their efficiency vary depending on the systems.

3.3 Solving the higher-order differential equation at order k

Now, in order to solve the HODE, we need one crucial assumption: it must be **first-order factorizable**, i.e. there must exist rational functions $\hat{p}_i(x) \in \mathbb{K}(x)$ such that we can write the homogeneous HODE in the following form:

$$(\mathrm{d}_x - \hat{p}_m(x))(\mathrm{d}_x - \hat{p}_{m-1}(x)) \cdots (\mathrm{d}_x - \hat{p}_1(x)) y(x) = 0. \quad (8)$$

Then, solutions of the HODE respecting such a factorization in a sense that we precise below are called **d'Alembertian solutions** [12].
Note: There exist algorithms to check whether the HODE is factorizable and to perform it whenever it is possible [23].

Supposing that we are able to do so, we define then the following hyperexponential functions that are the solutions of the first-order factors of the HODE:

$$\forall k \in \{1, \ldots, m\},\ h_k(x) = e^{\int_{\ell'_k}^{x} \mathrm{d}y \hat{p}_k(y)} \Leftrightarrow (\mathrm{d}_x - \tilde{p}_k(x))h_k(x) = 0 \text{ and } \ell'_k \in \mathbb{K}. \tag{9}$$

One can check that the m homogeneous d'Alembertian solutions are the following ones:

$$y_1(x) = h_1(x),$$
$$y_2(x) = h_1(x) \int_{\ell_0}^{x} \mathrm{d}x_1 \frac{h_2(x_1)}{h_1(x_1)},\ \ell_0 \in \mathbb{K},$$
$$\vdots$$
$$y_i(x) = h_1(x) \int_{\ell_0}^{x} \mathrm{d}x_1 \frac{h_2(x_1)}{h_1(x_1)} \cdots \int_{\ell_{i-2}}^{x_{i-2}} \mathrm{d}x_{i-1} \frac{h_i(x_{i-1})}{h_{i-1}(x_{i-1})},\ \ell_{i-2} \in \mathbb{K},$$
$$\vdots$$
$$y_m(x) = h_1(x) \int_{\ell_0}^{x} \mathrm{d}x_1 \frac{h_2(x_1)}{h_1(x_1)} \cdots \int_{\ell_{m-2}}^{x_{m-2}} \mathrm{d}x_{m-1} \frac{h_m(x_{m-1})}{h_{m-1}(x_{m-1})},\ \ell_{m-2} \in \mathbb{K}.$$

The solution vector space is therefore:

$$\mathcal{S}_h = \left\{ \sum_{i=1}^{m} c_i y_i(x) \middle| c_i \in \mathbb{K} \right\}.$$

The d'Alembertian solutions above are constructed iteratively using the following idea[6]: suppose that we have a simple solution for our equation (8) – and this is indeed the case, taking $y(x) = h_1(x)$ that cancels the first factor. Now, we define the ansatz $y(x) = h_1(x) \int^x \mathrm{d}x_1\, u(x_1)$, for some $u(x)$ that we want to find. Plugging this in (8) gives us:

$$\begin{aligned} \text{LHS}\,(8) &= (\mathrm{d}_x - \hat{p}_m(x)) \cdots (\mathrm{d}_x - \hat{p}_2(x)) \left[\mathrm{d}_x \left(h_1(x) \int^x \mathrm{d}x_1\, u(x_1) \right) - \hat{p}_1(x) h_1(x) \int \mathrm{d}x_1\, u(x_1) \right] \\ &= (\mathrm{d}_x - \hat{p}_m(x)) \cdots (\mathrm{d}_x - \hat{p}_2(x)) \left[h_1(x) u(x) \right] \end{aligned}$$

[6]For more details, see for instance [12].

Since we know that $h_2(x)$ is a solution (up to a constant) of $(\mathrm{d}_x - \hat{p}_2(x))z(x) = 0$:

$$u(x)h_1(x) = h_2(x) \Leftrightarrow u(x) = \frac{h_2(x)}{h_1(x)},$$

we get the solution

$$y_2(x) = h_1(x) \int_{\ell_0}^{x} \mathrm{d}x_1 \frac{h_2(x_1)}{h_1(x_1)}.$$

It suffices to apply this procedure recursively to get all the other solutions $y_i(x)$, with $i \in \{3, \cdots, m\}$. Note that by doing this we prove the unicity of such solutions, but to be rigourous one would also need to prove their independence – this is less trivial, and done in detail in [12].

From the algorithmic side, given that the HODE factors in first-order factors, the function `SolveDE[diff_eqn_in_f,f[x],x]` of the package `HarmonicSums` of J. Ablinger [1, 2] will find the full set of solutions described above with all possible simplifications, in particular down to expressions involving base harmonic polylogarithms (HPL) and cyclotomic harmonic polylogarithms (CHPL)[7].

Step 3: Factorize the HODE and solve it using `HarmonicSums`' `SolveDE` function.

Note: It might very well happen that the HODE doesn't fully factor into only first-order factors, and some higher-order factors remain. Then, the system might still be solvable in the same way, provided that we manage to find enough solutions of the higher-order factor. For example, suppose that we have a third order system that factored into the following form:

$$(\mathrm{d}_x^2 + \hat{p}_{2,1}(x)\mathrm{d}_x + \hat{p}_{2,2}(x))(\mathrm{d}x - \hat{p}_1(x))y(x) = 0. \tag{10}$$

Then, supposing that one can solve with `SolveDE` [8] or a different solver, say the standard Mathematica `DSolve`, the higher-order factor and get all the solutions, e.g. here two, the solutions will have mostly the same form. More precisely, let us take for example $\hat{p}_{2,1}(x) = 1$, $\hat{p}_{2,2}(x) = -x$ and $\hat{p}_1(x) = \frac{x}{1+x^2}$. While `SolveDE` fails to find a solution to $(\mathrm{d}_x^2 + \mathrm{d}_x - x)y(x) = 0$, `DSolve` finds a two-dimensional solution space made up of Airy functions of the first and second kind:

[7]Where all the (C)HPL were maximally simplified using identities, shuffle and stuffle algebra of the associated S-sums for instance, see for instance [11].

[8]The Kovavacic's algorithm [31] is actually implemented in `SolveDE` [1] and can already find solutions outside of d'Alembertian ones for second-order factors.

```
In[1]:=   DSolve[D[y[x],{x,2}]+D[y[x],x]-x*y[x]==0,y[x],x]

Out[1]=   {{y[x]→c₁e^(-x/2) AiryAi[1/4+x] + c₂e^(-x/2) AiryBi[1/4+x]}}
```

So defining the following solutions:

$$h_{2,1}(x) = e^{-\frac{x}{2}}\mathrm{Ai}\left(x+\frac{1}{4}\right),\ h_{2,2}(x) = e^{-\frac{x}{2}}\mathrm{Bi}\left(x+\frac{1}{4}\right),$$

the solutions of the whole HODE are therefore:

$$y_1(x) = h_1(x) = e^{\int^x \mathrm{d}y\, \frac{x}{1+x^2}} = \sqrt{1+x^2},$$

$$y_2(x) = h_1(x)\int_{\ell_{0,1}}^x \mathrm{d}y\, \frac{h_{2,1}(x)}{h_1(x)} = \sqrt{1+x^2}\int_{\ell_{0,1}}^x \mathrm{d}y\, \frac{e^{-\frac{y}{2}}\mathrm{Ai}\left(y+\frac{1}{4}\right)}{\sqrt{1+y^2}},$$

$$y_3(x) = h_1(x)\int_{\ell_{0,2}}^x \mathrm{d}y\, \frac{h_{2,2}(x)}{h_1(x)} = \sqrt{1+x^2}\int_{\ell_{0,2}}^x \mathrm{d}y\, \frac{e^{-\frac{y}{2}}\mathrm{Bi}\left(y+\frac{1}{4}\right)}{\sqrt{1+y^2}}.$$

Related to that, a rather general machinery has been developed in [1, 5, 8, 20] that will be explored further in the future.
In the same way as for the homogeneous solutions, one can actually check that the particular solution of the first equation of (7) has the following form:

$$g(x) = h_1(x)\int_{\ell_0}^x \mathrm{d}x_1 \frac{h_2(x_1)}{h_1(x_1)}\cdots\int_{\ell_{m-2}}^{x_{m-2}} \mathrm{d}x_{m-1}\frac{h_m(x_{m-1})}{h_{m-1}(x_{m-1})}\int_{\ell_{m-1}}^{x_{m-1}} \mathrm{d}x_m \frac{r(x_m)}{h_m(x_m)}. \quad (11)$$

So that combining the particular and homogeneous solutions, the general solution has the following form:

$$\boxed{\tilde{\mathcal{I}}_1^{(-l)}(x) = g(x) + c_1 y_1(x) + \cdots + c_m y_m(x) \text{ with } \forall i \in \{1,\ldots,m\},\ c_i \in \mathbb{K}.} \quad (12)$$

Finally, using the physical data, e.g. the value of the MI in several particular regimes of x, we can carry out

Step 4: Fix the general solution constants using the initial conditions.

3.4 Solving the system at order k and iterating

With this, the most technical part of the algorithm is done, since we have an explicit solution of the $(-l)$-th coefficient of our first MI, and all that remains is to unroll everything and get the other solutions. More particularly, we plug the solution (12) into the relations (7) relating the first MI to the others to get all those at order $(-l)$, i.e. for all $k \in \{2, \dots, m\}$:

$$\begin{aligned}\tilde{\mathcal{I}}_k^{(-l)}(x) &= \sum_{i=0}^{m-1} a_{k,i}(x) \mathrm{d}_x^i \tilde{\mathcal{I}}_1^{(-l)}(x) + \rho_k(x) \\ &= \sum_{i=0}^{m-1} \sum_{j=1}^{m} c_j a_{k,i}(x) \mathrm{d}_x^i y_j(x) + \sum_{i=0}^{m-1} a_{k,i}(x) g(x) + \rho_k(x).\end{aligned}$$

Finally, we can plug the whole vector of solution $\tilde{\mathcal{I}}^{(-l)} = (\tilde{\mathcal{I}}_1^{(-l)}, \dots, \tilde{\mathcal{I}}_n^{(-l)})$ into

$$\frac{d}{dx}\tilde{\mathcal{I}}^{(k)} = \tilde{\mathcal{M}}^{(0)}\tilde{\mathcal{I}}^{(k)} + \left(\sum_{i=1}^{k+l} \tilde{\mathcal{M}}^{(i)}\tilde{\mathcal{I}}^{(k-i)}\right) + \tilde{\mathcal{R}}^{(k)} \text{ with } k = -l+1. \tag{13}$$

Step 5: We reiterate the whole process, solving now the system (13).

Remark: Let us note that one of the advantages of the method presented above is that we don't have to redo all the steps at each iteration. Indeed, when we do the uncoupling, we can replace the explicit inhomogeneous part $\tilde{\mathcal{R}}^{(-l)}$ by a general vector of undefined functions, e.g. `Inhomog[(-l)]={R[1][x],...,R[n][x]}` in Mathematica. The uncoupling is in this case absolutely generic and depends only on the matrix coefficient. Since the latter is actually always $\tilde{\mathcal{M}}^{(0)}$ as seen in (5), the uncoupling doesn't change and so do the homogeneous solutions of the HODE. The only quantity that changes is the particular solution (11), since it depends directly on the inhomogeneous part of the system, which changes at each iteration.

4 Solving iteratively the system using the Sigma package

The first method presented above to solve iteratively the system at each order of ϵ concentrates on the series of subsystems obtained for each order of ϵ, that is solved iteratively by uncoupling each of them as first-order systems in one variable. There exists another method, presented in great detail in [45], that exchanges the order of step 1 and step 2, i.e. that uncouples the equation before solving it. Let us present it now.

4.1 Uncoupling the system

We suppose once more that the system has been triangularized before, and we are examining one of the 'tilde' sub-systems of the form (3) obtained as a consequence:

$$\frac{\mathrm{d}}{\mathrm{d}x}\tilde{\mathcal{I}}(x,\epsilon) = \tilde{\mathcal{M}}(x,\epsilon)\tilde{\mathcal{I}}(x,\epsilon) + \tilde{\mathcal{R}}(x,\epsilon).$$

In addition we assume that $\tilde{M}(x,\epsilon)$ is invertible, and that the inhomogeneous part can be expanded as a power series in x, i.e. for each $\tilde{R}_i(x,\epsilon)$ we can write:

$$\tilde{\mathcal{R}}_i(x,\epsilon) = \sum_{n=0}^{\infty} \tilde{R}_i(n,\epsilon)x^n. \tag{14}$$

Furthermore, we suppose that the Taylor coefficients themselves can be Laurent expanded in ϵ, i.e.

$$\tilde{R}_i(n,\epsilon) = \sum_{k=l}^{\infty} \tilde{R}_{i,k}(n)\epsilon^k \text{ with } l \in \mathbb{Z}. \tag{15}$$

We proceed now to the first alternative step[9]:

Step 2.b: Uncouple the equation (3) using `OreSys`.

We obtain a scalar HODE in x in one of the MI, say $\tilde{\mathcal{I}}(x)$:

$$\boxed{\alpha_0(x,\epsilon)\tilde{\mathcal{I}}_1(x,\epsilon) + \cdots + \alpha_m(x,\epsilon)\mathrm{d}_x^m\tilde{\mathcal{I}}_1(x,\epsilon) = \beta(x,\epsilon)} \tag{16}$$

where the $\alpha_i(x,\epsilon) \in \mathbb{K}[x,\epsilon]$. If we find those in the rational field $\mathbb{K}(x,\epsilon)$ instead of the ring, it suffices to multiply the whole equation by $\mathrm{lcm}(\mathrm{den}\,\alpha_1,\ldots,\mathrm{den}\,\alpha_n)$[10]. Similarly to the previous case, we have the inhomogeneous right-hand side (RHS) $r(x,\epsilon)$ of the equation that depends only on the inhomogeneous part of the system

[9]We will denote with a "b" the steps of this second approach.

[10]Where $\mathrm{den}\,\alpha_i$ denotes the denominator of α_i, i.e. if $\alpha_i(x,\epsilon) = \frac{p_i(x,\epsilon)}{q_i(x,\epsilon)}$ with $(p_i,q_i) \in \mathbb{K}[x,\epsilon] \times \mathbb{K}[x,\epsilon]^\star$, then $\mathrm{den}\,\alpha_i(x,\epsilon) = q_i(x,\epsilon)$.

and its derivatives, and the other MIs on the first MI, its derivatives and the RHS part, i.e.:

$$\beta(x,\epsilon) = \sum_{i=0}^{p}\sum_{j=1}^{m} \beta_{i,j}(x,\epsilon)\mathrm{d}_x^i\tilde{\mathcal{R}}_j(x,\epsilon) \text{ with } p \in \mathbb{N},\ \beta_{i,j}(x,\epsilon) \in \mathbb{K}(x,\epsilon), \tag{17}$$

$$\tilde{\mathcal{I}}_j(x,\epsilon) = \sum_{i=0}^{m-1} a_{k,i}(x,\epsilon)\mathrm{d}_x^i\tilde{\mathcal{I}}_1(x,\epsilon) + \rho_k(x,\epsilon) \text{ with } \ j \in \{2,\dots,m\},\ a_{k,i}(x) \in \mathbb{K}(x,\epsilon),$$
$$\rho_k(x,\epsilon) = \sum_{i=0}^{q}\sum_{j=1}^{m} \rho_{k,i,j}(x,\epsilon)\mathrm{d}_x^i\tilde{\mathcal{R}}_j(x,\epsilon) \text{ with } q \in \mathbb{N},\ \rho_{k,i,j}(x,\epsilon) \in \mathbb{K}(x,\epsilon). \tag{18}$$

Note: It is actually possible at this point to introduce a small refinement [45] that might be very helpful to improve the speed of our algorithm. Indeed, let us define

$$p(x) := \gcd(\alpha_0(x,0),\dots,\alpha_m(x,0))$$

Then we can divide the equation (16) by $p(x)$, so that

$$\sum_{n=0}^{m} \frac{\alpha_n(x,\epsilon)}{p(x)}\tilde{\mathcal{I}}_1(x,\epsilon) = \frac{\beta(x,\epsilon)}{p(x)}.$$

It is equivalent to (16) modulo the fact that the coefficients have a reduced degree and that the cofficients of the expansion of the RHS in ϵ changes.

4.2 Order by order solving

Now, we write the following ansatz for the MI $\tilde{\mathcal{I}}_1(x,\epsilon)$:

$$\tilde{\mathcal{I}}_1(x,\epsilon) = \sum_{n=0}^{\infty} I_1(n,\epsilon)x^n, \tag{19}$$

i.e. we suppose that it is regular in x. This leads us to the next step:

Step 3.b: Plugging the relation (19) in the HODE (16), we perform coeffcient comparison with respect to powers of x and write a recurrence of the form

$$a_0(n,\epsilon)I_1(n,\epsilon) + a_1(n,\epsilon)I_1(n+1,\epsilon) + \cdots + a_d(n,\epsilon)I_1(n+d,\epsilon) = b(n,\epsilon) \tag{20}$$

where $d \in \mathbb{N}$, $a_i(n,\epsilon) \in \mathbb{K}[n,\epsilon]$, $b(n,\epsilon)$ can be expanded in Laurent series

$$b(n,\epsilon) = \sum_{k=-l}^{\infty} b_k(n)\epsilon^k \text{ with } l \in \mathbb{N}. \tag{21}$$

In addition, we can compute the coefficients $b_{-l}(n), \ldots, b_r(n) \in \mathbb{K}(n)$, $r \in \mathbb{Z}$ for $n \in \mathbb{N}$ as expressions in terms of indefinite nested sums.

Proof. Let us show this for the homogeneous HODE[11], i.e. for $r(x,\epsilon) = 0$ and therefore $b(n,\epsilon) = 0$. For all $i \in \{1,\ldots,m\}$, since $\alpha_i(x,\epsilon) \in \mathbb{K}[x,\epsilon]$ we have

$$\alpha_i(x,\epsilon) = \sum_{r=0}^{p_i} \bar{a}_i(r,\epsilon)x^r \text{ with } p_i = \deg_x(\alpha_i) \in \mathbb{N}.$$

Therefore, using the Cauchy product:

$$\begin{aligned}
\alpha_i(x,\epsilon)\mathrm{d}_x^i\tilde{\mathcal{I}}_1(x,\epsilon) &= \left(\sum_{r=0}^{p_i} \bar{a}_i(r,\epsilon)x^r\right)\left(\sum_{n=i}^{\infty} \frac{n!}{(n-i)!}I_1(n,\epsilon)x^{n-i}\right) \\
&= \left(\sum_{r=0}^{p_i} \bar{a}_i(r,\epsilon)x^r\right)\left(\sum_{n=0}^{\infty} \frac{(n+i)!}{n!}I_1(n+i,\epsilon)x^{n}\right) \\
&= \sum_{n=0}^{\infty} C_i(n,\epsilon)x^n
\end{aligned}$$

where, since for all $n > p_i$ we have $a_i(n,\epsilon) = 0$, the Cauchy product $C_i(n,\epsilon)$ can be written as[12]

$$C_i(n,\epsilon) := \sum_{k=0}^{\min(n,p_i)} \frac{(n+i-k)!}{(n-k)!}\bar{a}_i(k,\epsilon)I_1(n+i-k,\epsilon)$$

We plug this in (16) and we get:

$$\alpha_0(x,\epsilon)\tilde{\mathcal{I}}_1(x,\epsilon) + \cdots + \alpha_m(x,\epsilon)\mathrm{d}_x^m\tilde{\mathcal{I}}_1(x,\epsilon) = 0 \Rightarrow \sum_{n=0}^{\infty}\left[\sum_{i=0}^{m} C_i(n,\epsilon)\right]x^n = 0$$

That is to say we have for $n \geq \max_{1\leq i\leq m} p_i$:

$$\boxed{\sum_{i=0}^{m}\sum_{k=0}^{p_i} \frac{(n+i-k)!}{(n-k)!}\bar{a}_i(k,\epsilon)I_1(n+i-k,\epsilon) = 0} \tag{22}$$

[11]For more informations on holonomic fonctions and sequences, see for instance [34].

[12]This operation is perfectly fine within the ring of formal power series. If one wants to stay in the analytic world, one might need to check the absolute convergence of the $\mathrm{d}_x^i\tilde{\mathcal{I}}_1(x,\epsilon)$ series in order to perform the Cauchy product on it and $\alpha_i(x,\epsilon)$.

We define now

$$M := \max_{1 \leq i \leq m} p_i.$$

The coefficient in front of $I_1(n+j,\epsilon)$ for $j \in \{-M,\dots,-1,0,1,\dots,m\}$ is obtained by setting $i-k=j \Leftrightarrow i=k+j$ and collecting the relevant $\bar{a}_i(k,\epsilon)$ coefficients, i.e. by defining

$$\tilde{a}_j(n,\epsilon) := \sum_{r=\max(0,-j)}^{m-j} \frac{(n+j)!}{(n-r)!} \bar{a}_{r+j}(r,\epsilon). \tag{23}$$

We get a difference equation of the following form:

$$\tilde{a}_{-M}(n,\epsilon) I_1(n-M,\epsilon) + \dots + a_m(n,\epsilon) I_1(n+m,\epsilon) = 0. \tag{24}$$

Shifting everything by M and defining the following quantities:

$$\boxed{a_j(n,\epsilon) := \tilde{a}_{j-M}(n,\epsilon) = \sum_{r=\max(0,M-j)}^{m+M-j} \frac{(n+j-M)!}{(n-r)!} \bar{a}_{r+j-M}(r,\epsilon)} \tag{25}$$

we finally get the expected difference equation:

$$a_0(n,\epsilon) I_1(n,\epsilon) + \dots + a_d(n,\epsilon) I_1(n,\epsilon) = 0; \tag{26}$$

it is of order $d \leq m+M$, since some of the $a_j(n,\epsilon)$ might be equal to 0 or cancel among themselves. $\square$

Note: With the refinement introduced above, consisting in dividing (16) by the GCD of the coefficients, there is actually an even stronger bound on d, that is to say $d \leq m + M - \deg p$.

At this point, we obtained a difference equation of higher order in n that still depends on ϵ. In order to solve it, we need to get rid of the latter.

Step 4.b: We solve the difference equation (26) order by order in ϵ, i.e. we search for a Laurent expansion for $I_1(n,\epsilon)$ of the form

$$I_1(n,\epsilon) = \sum_{k=-l}^{\infty} I_{1,k}(n)\epsilon^k \tag{27}$$

where $l \in \mathbb{N}$ and the $I_{1,k}(n)$ can be given in terms of indefinite nested sums over hypergeometric products.

Definition 4.1 (hypergeometric sequence)**.** *A sequence $(f(n))_{n\geq 0}$ is called **hypergeometric** if $f(n) \neq 0\ \forall n \geq l$ for some $l \in \mathbb{N}$ and there exists a rational function $r(x) \in \mathbb{K}(x)$ such that*

$$\forall n \geq l,\ \frac{f(n+1)}{f(n)} = r(n).$$

Note: $(f(n))_{n\geq 0}$ with $f(x) \in \mathbb{K}(x)$ without poles is an hypergeometric sequence.

Often hypergeometric sequences can be represented by hypergeometric products:

Definition 4.2 (hypergeometric product)**.** *Let $f(x) \in \mathbb{K}(x)$ and $l, n \in \mathbb{N}$. If $\forall j \geq l,\ f(j) \neq 0$ and its evaluation does not introduce poles, then $\prod_{j=l}^{n} f(j)$ is called an **hypergeometric product**.*

Examples:

- $n! = \prod_{i=1}^{n} i$
- $\binom{m}{n} = \prod_{i=1}^{n} \frac{m+1-i}{i}$

Definition 4.3 (indefinite nested sums over HGP [45])**.** *An **expression in terms of indefinite nested sums over hypergeometric products (INSH)** in k over $\mathbb{K}$ is composed recursively by the three operations $(+,-,\cdot)$ with:*

- *elements from the rational function field $\mathbb{K}(k)$,*
- *hypergeometric products in k over $\mathbb{K}$,*
- *and sums of the form $\sum_{j=l}^{k} f(j)$ with $l \in \mathbb{N}$ where $f(j)$ is an expression of INSH in j over $\mathbb{K}$*[13]*.*

[13]Here it is assumed that the evaluation of $f(j)|_{j\mapsto m}$ for all $m \in \mathbb{Z}$ with $m \geq l$ does not introduce any poles.

That is to say, after the use of the quasi-shuffle algebra [11], it simplifies down to an expression of the form:

$$S(n) = \sum_{i_1=l_0}^{n} f_1(i_1) \sum_{i_2=l_1}^{i_1} f_2(i_2) \cdots \sum_{i_n=l_{n-1}}^{i_{n-1}} f_n(i_n)$$

where the $f_1, \ldots, f_n$ are hypergeometric products. If such an expression is a solution to a linear difference equation, it is called a ***d'Alembertian solution*** *[12].*

In order to solve the difference equation (26) recursively, we suppose first that all the $a_i(x, 0) \in \mathbb{K}[x, \epsilon]$ are not zero at the same time, which can always be achieved. Indeed, if this is the case, we can write them all in the form:

$$a_i(x, \epsilon) = \epsilon^{u_i} \bar{a}_i(x, \epsilon) \text{ where } u_i \in \mathbb{N},\ \bar{a}_i(x, \epsilon) \in \mathbb{K}[x, \epsilon],\ \bar{a}_i(x, 0) \neq 0.$$

Then by setting $u := \operatorname{lcm}(u_1, \ldots, u_d)$ we divide (26) by ϵ^u so that the equation becomes:

$$\bar{a}_0(n, \epsilon) I_1(n, \epsilon) + \cdots + \bar{a}_d(n, \epsilon) I_d(n, \epsilon) = \bar{b}(n, \epsilon)$$

where the lowest pole of $b(n, \epsilon)$ around $\epsilon = 0$ is shifted by u:

$$\bar{b}(n, \epsilon) := \sum_{k=-l}^{\infty} b_k(n) \epsilon^{k-u} = \sum_{k=-(l+u)}^{\infty} b_{k+u}(n) \epsilon^k.$$

Now, we can define a maximal $d' \in \mathbb{N}$, $d' < d$ such that $a_{d'}(x, 0) \neq 0$ as well as a $M \in \mathbb{N}$ such that for all $n \geq M$, $a_{d'}(n, 0) \neq 0$. The idea is to algorithmically construct when possible an expansion of the form (27) using `Sigma`[14], with given initial values $I_{1,j}(i) = c_{i,j}$ for $j \in \{l, \ldots, r\}$ and $i \in \{0, \ldots, \max(d', M) - 1\}$. The general steps of the method are as follows:

1. We plug the expansion (27) in (26) :

$$\begin{aligned}
&a_0(n, \epsilon) \left[I_{1,-l}(n) \epsilon^{-l} + I_{-l+1,1}(n) \epsilon^{-l+1} + \cdots \right] \\
&+ a_1(n, \epsilon) \left[I_{1,-l}(n+1) \epsilon^{-l} + I_{-l+1,1}(n+1) \epsilon^{-l+1} + \cdots \right] \\
&+ \cdots \\
&+ a_{d'}(n, \epsilon) \left[I_{1,-l}(n+d) \epsilon^{-l} + I_{-l+1,1}(n+d') \epsilon^{-l+1} + \cdots \right] = b_{-l}(n) \epsilon^{-l} + b_{-l+1}(n) \epsilon^{-l+1}
\end{aligned}$$

[14]See for more detail [19].

and then collect terms:

$$
\begin{aligned}
&[a_0(n,\epsilon)I_{1,-l}(n)+\cdots+a_{d'}(n,\epsilon)I_{1,-l}(n+d')-b_l(n)]\,\epsilon^{-l}\\
+&[a_0(n,\epsilon)I_{1,-l+1}(n)+\cdots+a_{d'}(n,\epsilon)I_{1,-l+1}(n+d')-b_{-l+1}(n)]\,\epsilon^{-l+1}\\
+&\cdots=0.
\end{aligned}
$$

2. We multiply everything by ϵ^l and then send $\epsilon \to 0$. Since $a_i(n,0)\neq 0$, the only term that remains is therefore

$$a_0(n,0)I_{1,-l}(n)+\cdots+a_{d'}(n,0)I_{-l,1}(n+d')=b_l(n).$$

3. Now we use `Sigma` to solve[15] the obtained recurrence equation and to combine the solutions such that the initial conditions are fulfilled. If `SolveRecurrence` fails to find a solution, then we stop. Otherwise, we obtain an expression of $I_{1,-l}(n)$ in terms of INSH.

4. We define then

$$I_1'(n,\epsilon):=I_1(n,\epsilon)-I_{1,-l}(n)\epsilon^{-l}=\sum_{k=-l+1}^{\infty}I_{1,k}(n)\epsilon^k \tag{28}$$

and plug this into (26):

$$a_0(n,\epsilon)\left[I_{1,l}(n)\epsilon^l+I_1'(n,\epsilon)\right]+\cdots+a_{d'}(n,\epsilon)\left[I_{1,l}(n+d')\epsilon^l+I_1'(n+d')\right]=b_l(n,\epsilon).$$

5. Collecting the ϵ^l terms and getting them on the other side, we get an equation of a form similar to (26) for $I_1'(n,\epsilon)$ with an updated RHS:

$$a_0(n,\epsilon)I_1'(n,\epsilon)+\cdots+a_{d'}(n,\epsilon)I_1'(n+d',\epsilon)=b'(n,\epsilon) \tag{29}$$

where

$$b'(n,\epsilon):=b(n,\epsilon)-[a_0(n,\epsilon)I_{1,-l}(n)+\cdots+a_{d'}(n,\epsilon)I_{1,-l}(n+d',\epsilon)]\,\epsilon^{-l}.$$

Note that the coefficients of the pole of order l in $b'(n,\epsilon)$ cancel among themselves and $b'(n,\epsilon)$ has only a pole of order $l-1$ around $\epsilon=0$.

6. We loop this procedure to get the next term $I_{1,-l+1}(n)$.

[15]The underlying algorithm can be considered as the discrete version described in section 3.3. In particular, we have to assume that the recurrence factorizes into first-order factors (see [12]).

4.3 Solving for the other master integrals and iterating

Step 5.b: Once we get a solution for $\tilde{\mathcal{I}}_1(x,\epsilon)$ up to some order l' in ϵ, we plug it in (18) in order to get the other MIs up to the required order in ϵ.

Note: It might well be possible that due to the structure of the equations (18), there are some simplifications or some coefficients $a_{k,i}(x,\epsilon)$ that have poles in ϵ. What is actually necessary to do is some additional preprocessing in order to find the order $k \in \mathbb{N}$ of the maximal pole in each of those expressions for the $\tilde{\mathcal{I}}_j(x,\epsilon)$ and compute $\tilde{\mathcal{I}}_1(x,\epsilon)$ up to the required order$+k$.

4.4 Alternative application of the method

The second method presented above relies, compared to the first one, on the setting of recurrences and `Sigma`, whereas the first one was really dealing with differential equations through extensive use of the `SolveDE` function of the `HarmonicSums` package. It is actually possible to remain in this setting also for the second method, and this consists in taking the following twist[16]: after the Step 2.b, i.e. the uncoupling through `OreSys` of the whole subsystem, instead of expanding $\tilde{\mathcal{I}}_1(x,\epsilon)$ in x, we expand in ϵ, i.e.:

$$\tilde{\mathcal{I}}_1(x,\epsilon) = \sum_{n=0}^{\infty} I_1(n,\epsilon)x^n \to \tilde{\mathcal{I}}_1(x,\epsilon) = \sum_{k=-l}^{\infty} \tilde{\mathcal{I}}_{1,k}(x)\epsilon^k \tag{30}$$

Note: Actually, to be once again perfectly rigourous and if one does not want to stay within the formal Laurent setting, one might need to use some discrete version of the Fubini theorem to check whether this alternative expansion is equivalent to the first one, i.e. if one is allowed to interchange freely the ϵ and x sums:

$$\tilde{\mathcal{I}}_1(x,\epsilon) = \sum_{n=0}^{\infty}\sum_{k=-l}^{\infty} I_1(n)\epsilon^k x^n = \sum_{k=-l}^{\infty}\sum_{n=0}^{\infty} I_1(n)\epsilon^k x^n$$

Now, expanding the $\alpha_i(x,\epsilon)$ and $\beta(x,\epsilon)$ of (16) in ϵ instead of x, we can proceed exactly in the spirit of the ideas of the proof for the step 3.b as well as those of step 4.b: plug all the ϵ expansions in the HODE (16), identify terms with same order of ϵ and then send ϵ to 0 to get a HODE purely in x of the following form for the first coefficient $\tilde{\mathcal{I}}_1(x,\epsilon)$:

$$a_0(x,0)\tilde{\mathcal{I}}_1^{(-l)}(x) + \cdots + a_d(x,0)\mathrm{d}_x^d\tilde{\mathcal{I}}_1^{(-l)}(x) = \beta_{-l}(x) \tag{31}$$

[16]Compare also [8].

where $d \leq m$ and the $a_i(x,0) \in \mathbb{K}[x]$ are obtained from linear combinations of the ϵ-expansion coefficients of the $\alpha_i(x,\epsilon)$, in the spirit of (25).
Here we can apply directly `SolveDE` in order to get the homogeneous and particular solutions of the HODE, given as always that it is first-order factorizable, and using the initial conditions, we solve completely for $\tilde{\mathcal{I}}_{1,-l}(x)$. The rest of the computation proceeds in the same way [45], i.e. plug the solution in the initial HODE (16) and repeat until the required order is reached. Finally, we insert the obtained solution in the uncoupling equation relating the $\tilde{\mathcal{I}}_j(x,\epsilon)$ to $\tilde{\mathcal{I}}_1(x,\epsilon)$.

5 Extending the scope of the algorithm

The algorithms presented above allow one to solve efficiently different physical problems. In particular, the first one has been used to compute the MIs that contribute to both the color-planar and complete light quark non-singlet three-loop contributions to the heavy-quark form factors for different currents [10]. The second one has been for instance applied to compute polarized three-loop anomalous dimensions [13] as well as heavy fermion contributions of the massive three loop form factors [20]. Nevertheless, there remain several aspects that will constitute our future line of work to improve the algorithms and their implementation:

- For the first method, we required in subsection 2.1 that in the matrix expansion $\mathcal{M}(x,\epsilon) = \sum_{i=0}^{\infty} \mathcal{M}^{(i)}(x)\epsilon^i$, the determinant of the first coefficient is non-zero. Otherwise, the uncoupling in step 2 will fail. One of the tasks will consist to find a way to overcome this shortcoming, possibly by doing some additional triangularization step/Gaußian elimination and extracting the maximally non-degenerate matrix to uncouple smaller subsystems.

- In the second note of subsection 3.3, we point out the fact that it might happen that the HODE is not completely first-order factorizable. Due to the Kovacic algorithm [31] implementation in the package `HarmonicSums` [1], `SolveDE` actually can find solutions exceeding the class of d'Alembertian solutions, in particular Liovillian solutions, in case of second-order factors remaining, but equations with solutions such as the ones presented in the second note of subsection 3.3 are still out of scope at the moment. A similar issue exists for the discrete case, that also needs to be extended. Moreover, it might happen that even in case such a solution is found for a second-order factor, the remaining IOH might be quite complex, so it is also necessary to find a way to obtain the simplest alphabet for the homogeneous solutions.

- During the uncoupling step, several algorithms are available – in particular Zürcher [16, 22, 46], Gauß and Euclid algorithms, that are more or less efficient depending on the system. Moreover, even after triangularization, some systems are still relatively big (of order ~ 10) and the uncoupling step with `OreSys` can be quite slow. It is thus relevant to write a preprocessing part that would identify automatically the fastest uncoupling scheme when given a certain system and to find the best uncoupling by cycling through the system variables, and if possible to improve the uncoupling algorithms themselves in a second step.

- Until now, in particular for the first method, several manual inputs, sometimes at different levels of the processing, are necessary in order to run the algorithms – we have to input directly the most reduced system after the triangularization step, and fine-tune the initial conditions by hand. Therefore, it appears necessary to implement a full pre-processing step. Given a certain system and the order up to which we want to solve it, it should subdivide the system into the smallest possible subsystems in a triangular form, and return all the relevant information that the user needs to input directly at once (for instance, order of expansion of the BIs). In this way, the uncoupling method should work completely automatically without any further control of the user.

- The advantage of the latter method is that it can be used to implement the large momentum method [21], implemented in the packages `SolveCoupledSystems` [4][3][7][15], which computes the coefficients $I_1(n, \epsilon)$ appearing in (19) for $1 \leq n \leq s$, where usually $s \sim 10^3 - 10^4$. The idea is also to implement the large moment method that can be called as an alternative when the HODE doesn't factor to first or second order factors, so that we can still constrain the considered MIs.

In general, all the above proposals will be implemented while taking into account complexity issues in order to enhance and speed up as much as possible the underlying algorithms.

Currently, what has already been carried over is the following:

- The first method has been completely implemented and tested for some systems, coming in particular from [10]. It has also been optimized so that computations are more efficient and are as fast as possible.

- A first version of an analysis algorithm has been implemented for a unique system[17]. Cycling through the MIs, it extracts the best uncoupling (with respect to the order of the HODE) as well as relevant information for the MIs (needed number of initial conditions) and the BIs (the required expansion in ϵ).

6 Conclusion

In this report, we presented two methods to solve first-order factorisable systems of differential equations. The first one [10] relies on an ϵ expansion of the original system and uncoupling of the obtained series of subsystems that need to be solved recursively. Using in particular the `HarmonicSums` package [2][11], this method has already been applied for higher-loop computations in QCD, for instance see [45][5][9][18]. The second method [45] consists in expanding directly order by order in x (respectively ϵ) the HODE and solving recursively the subsequently obtained difference equations using `Sigma` (respectively, the scalar differential equations using `SolveDE`). Currently, in order to circumvent some issues or to expand the scope of the algorithm as seen in the previous section 5, a completely automatic code is under development.

[17]It actually also tries first to separate the system into smaller independent clusters before analysing it.

References

[1] J. Ablinger. *Computing the Inverse Mellin Transform of Holonomic Sequences using Kovacic's Algorithm*. 2018. arXiv: `1801.01039 [cs.SC]`.

[2] J. Ablinger. *The package HarmonicSums: Computer Algebra and Analytic aspects of Nested Sums*. 2014. arXiv: `1407.6180 [cs.SC]`.

[3] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, A. von Manteuffel, and C. Schneider. "Calculating three loop ladder and V-topologies for massive operator matrix elements by computer algebra". In: *Computer Physics Communications* 202 (May 2016), pp. 33–112. ISSN: 0010-4655. DOI: `10.1016/j.cpc.2016.01.002`.

[4] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, and C. Schneider. *Algorithms to solve coupled systems of differential equations in terms of power series*. 2016. arXiv: `1608.05376 [cs.SC]`.

[5] J. Ablinger, A. Behring, J. Blümlein, G. Falcioni, A. De Freitas, P. Marquard, N. Rana, and C. Schneider. "Heavy quark form factors at two loops". In: *Physical Review D* 97.9 (May 2018). ISSN: 2470-0029. DOI: `10.1103/physrevd.97.094022`.

[6] J. Ablinger, A. Behring, J. Blümlein, A. De Freitas, E. Imamoglu, M. van Hoeij A. von Manteuffel, C.G. Raab, C.-S. Radu, and C. Schneider. "Iterative and Iterative-Noniterative Integral Solutions in 3-Loop Massive QCD Calculations". In: *Proc. of the 13th International Symposium on Radiative Corrections (Applications of Quantum Field Theory to Phenomenology)*. Ed. by A. Hoang and C. Schneider. Vol. PoS (RADCOR2017) 069. arXiv:1711.09742 [hep-ph]. 2018, pp. 1–13. URL: `https://doi.org/10.22323/1.290.0069`.

[7] J. Ablinger, J. Blümlein, A. De Freitas, and C. Schneider. *A toolbox to solve coupled systems of differential and difference equations*. 2016. arXiv: `1601.01856 [cs.SC]`.

[8] J. Ablinger, J. Blümlein, A. De Freitas, M. van Hoeij, E. Imamoglu, C.G. Raab, C.-S. Radu, and C. Schneider. "Iterated Elliptic and Hypergeometric Integrals for Feynman Diagrams". In: *J. Math. Phys.* 59.062305 (2018). arXiv:1706.01299 [hep-th],doi.org/10.1063/1.4986417, pp. 1–55. URL: `https://arxiv.org/abs/1706.01299`.

[9] J. Ablinger, J. Blümlein, A. Freitas, A. Hasselhuhn, A. Manteuffel, M. Round, C. Schneider, and F. Wißbrock. "The transition matrix element $A_{gq}(N)A_{gq}(N)$ of the variable flavor number scheme at $O(\alpha_s^3)$". In: *Nuclear Physics B* 882 (Feb. 2014), pp. 263–288. DOI: `10.1016/j.nuclphysb.2014.02.007`.

[10] J. Ablinger, J. Blümlein, P. Marquard, N. Rana, and C. Schneider. "Automated solution of first order factorizable systems of differential equations in one variable". In: *Nuclear Physics B* 939 (Dec. 2018). DOI: `10.1016/j.nuclphysb.2018.12.010`.

[11] J. Ablinger, J. Blümlein, and C. Schneider. "Analytic and algorithmic aspects of generalized harmonic sums and polylogarithms". In: *Journal of Mathematical Physics* 54.8 (Aug. 2013), p. 082301. ISSN: 1089-7658. DOI: `10.1063/1.4811117`. URL: `http://dx.doi.org/10.1063/1.4811117`.

[12] S.A. Abramov and M. Petkovšek. "D'Alembertian Solutions of Linear Differential and Difference Equations". In: (1994), pp. 169–174.

[13] A. Behring, J. Blümlein, A. De Freitas, A. Goedicke, S. Klein, A. von Manteuffel, C. Schneider, and K. Schönwald. "The polarized three-loop anomalous dimensions from on-shell massive operator matrix elements". In: *Nuclear Physics B* 948 (Nov. 2019), p. 114753. ISSN: 0550-3213. DOI: `10.1016/j.nuclphysb.2019.114753`.

[14] J. Blümlein. "Algebraic relations between harmonic sums and associated quantities". In: *Computer Physics Communications* 159.1 (May 2004), pp. 19–54. ISSN: 0010-4655. DOI: `10.1016/j.cpc.2003.12.004`.

[15] J. Blümlein, A. De Freitas, and C. Schneider. *Recent Symbolic Summation Methods to Solve Coupled Systems of Differential and Difference Equations*. 2014. arXiv: `1407.2537 [cs.SC]`.

[16] J. Blümlein, A. De Freitas, and C. Schneider. "Recent Symbolic Summation Methods to Solve Coupled Systems of Differential and Difference Equations". In: *PoS* LL2014 (2014), p. 017. DOI: `10.22323/1.211.0017`.

[17] J. Blümlein, A. De Freitas, M. van Hoeij, E. Imamoglu, P. Marquard, and C. Schneider. "The ρ parameter at three loops and elliptic integrals". In: *Proceedings of "Loops and Legs in Quantum Field Theory - LL 2018", 29 April - 4 May 2018*. Ed. by J. Blümlein and P. Marquard. PoS(LL2018)017. arXiv:1807.05287 [hep-ph]. 2018, pp. 1–14. URL: `https://doi.org/10.22323/1.303.0017`.

[18] J. Blümlein, A. Hasselhuhn, S. Klein, and C. Schneider. “The contributions to the gluonic massive operator matrix elements”. In: *Nuclear Physics B* 866.2 (Jan. 2013), pp. 196–211. ISSN: 0550-3213. DOI: 10.1016/j.nuclphysb.2012.09.001.

[19] J. Blümlein, S. Klein, C. Schneider, and F. Stan. “A symbolic summation approach to Feynman integral calculus”. In: *Journal of Symbolic Computation* 47.10 (Oct. 2012), pp. 1267–1289. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2011.12.044.

[20] J. Blümlein, P. Marquard, N. Rana, and C. Schneider. “The heavy fermion contributions to the massive three loop form factors”. In: *Nuclear Physics B* 949 (Dec. 2019), p. 114751. ISSN: 0550-3213. DOI: 10.1016/j.nuclphysb.2019.114751.

[21] J. Blümlein and C. Schneider. “The method of arbitrarily large moments to calculate single scale processes in quantum field theory”. In: *Physics Letters B* 771 (Aug. 2017), pp. 31–36. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2017.05.001.

[22] A. Bostan, F. Chyzak, and É. de Panafieu. “Complexity Estimates for Two Uncoupling Algorithms”. In: Boston, Maine, USA: Association for Computing Machinery, 2013. ISBN: 9781450320597. DOI: 10.1145/2465506.2465941.

[23] M. Bronstein. “Linear Ordinary Differential Equations: Breaking through the Order 2 Barrier”. In: *Papers from the International Symposium on Symbolic and Algebraic Computation*. ISSAC ’92. Berkeley, California, USA: Association for Computing Machinery, 1992, pp. 42–48. ISBN: 0897914899. DOI: 10.1145/143242.143264.

[24] K.G. Chetyrkin and F.V. Tkachov. “Integration by parts: The algorithm to calculate beta-functions in 4 loops”. In: *Nuclear Physics B* 192.1 (1981), pp. 159–204. ISSN: 0550-3213. DOI: 10.1016/0550-3213(81)90199-1.

[25] C.F. Gauss. *Theoria attractionis corporum sphaeroidicorum ellipticorum homogeneorum methodo nova tractata*. Königliche Gesellschaft der Wissenschaften, 1813.

[26] S. Gerhold. “Uncoupling systems of linear operator equations”. PhD thesis. RISC, J. Kepler University, Linz, Feb. 2002.

[27] G. Green. “Essay on the Mathematical Theory of Electricity and Magnetism”. In: *Essay on the Mathematical Theory of Electricity and Magnetism* (1828), pp. 1–115.

[28] J.M. Henn. "Multiloop Integrals in Dimensional Regularization Made Simple". In: *Phys. Rev. Lett.* 110 (25 June 2013), p. 251601. DOI: 10.1103/PhysRevLett.110.251601.

[29] M. Karr. "Summation in Finite Terms". In: *Journal of the ACM (JACM)* 28.2 (1981), pp. 305–350.

[30] A.V. Kotikov. "Differential equations method. New technique for massive Feynman diagram calculation". In: *Physics Letters B* 254.1 (1991), pp. 158–164. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(91)90413-K.

[31] J.J. Kovacic. "An algorithm for solving second order linear homogeneous differential equations". In: *Journal of Symbolic Computation* 2.1 (1986), pp. 3–43. ISSN: 0747-7171. DOI: https://doi.org/10.1016/S0747-7171(86)80010-4.

[32] J. Lagrange. "Nouvelles recherches sur la nature et la propagation du son, Miscellanea Taurinensis, t. II". In: *Oeuvres t.* 1 (1760), p. 263.

[33] S. Laporta. "High-precision calculation of multiloop Feynman integrals by difference equations". In: *International Journal of Modern Physics A* 15.32 (2000), pp. 5087–5159.

[34] C. Mallinger. "Algorithmic Manipulations and Transformations of Univariate Holonomic Functions and Sequences". PhD thesis. RISC, J. Kepler University, Linz, Aug. 1996.

[35] A. von Manteuffel and C. Studerus. *Reduze 2 - Distributed Feynman Integral Reduction.* 2012. arXiv: 1201.4330 [hep-ph].

[36] P. Marquard and D. Seidel. *The package Crusher.* unpublished.

[37] M. Ostrogradski. In: *Mem.Ac.Sci.St.Peters.* 6 (1831), pp. 129–133.

[38] M. van der Put and M. Singer. *Galois Theory of Linear Differential Equations.* Springer, 2003.

[39] E. Remiddi. "Differential equations for Feynman graph amplitudes". In: *Nuovo Cim. A* 110 (1997), pp. 1435–1452. arXiv: hep-th/9711188.

[40] C. Schneider. "A collection of denominator bounds to solve parameterized linear difference equations in ΠΣ-extensions". In: *An.Univ.Timişoara Ser.Mat.-Inform.* 42.2 (2004), pp. 163–179.

[41] C. Schneider. "Degree Bounds to Find Polynomial Solutions of Parameterized Linear Difference Equations in ΠΣ-Fields". In: *Applicable Algebra in Engineering, Communication and Computing* 16.1 (July 2005), pp. 1–32. ISSN: 1432-0622. DOI: 10.1007/s00200-004-0167-3.

[42] C. Schneider. “Simplifying sums in $\Pi\Sigma$-extensions”. In: *J.Algebra Appl.* 6.3 (2007), pp. 415–441.

[43] C. Schneider. “Solving parameterized linear difference equations in terms of indefinite nested sums and products”. In: *Journal of Difference Equations and Applications* 11.9 (2005), pp. 799–821. DOI: 10.1080/10236190500138262.

[44] C. Schneider. “Symbolic summation in difference fields”. PhD thesis. JKU Linz, 2001.

[45] C. Schneider, J. Blümlein, and P. Marquard. “A refined machinery to calculate large moments from coupled systems of linear differential equations”. In: *PoS* RADCOR2019 (2019), p. 078. DOI: 10.22323/1.375.0078.

[46] B. Zürcher. “Rationale Normalformen von pseudo-linearen Abbildungen”. MA thesis. ETH Zürich, 1994.