Published for SISSA by Springer

Received: *September 16, 2020*
Revised: *October 19, 2020*
Accepted: *October 24, 2020*
Published: *December 9, 2020*

IBP reduction coefficients made simple

JHEP12(2020)054

Janko Boehm,[a] Marcel Wittmann,[a] Zihao Wu,[b,c] Yingxuan Xu[d] and Yang Zhang[b,c,e]

[a] *Department of Mathematics, Technische Universität Kaiserslautern,*
67663 Kaiserslautern, Germany

[b] *Peng Huanwu Center for Fundamental Theory,*
Hefei, Anhui 230026, China

[c] *Interdisciplinary Center for Theoretical Study, University of Science and Technology of China,*
Hefei, Anhui 230026, China

[d] *ETH Zürich, Institut für Theoretische Physik,*
Wolfgang-Pauli-Str. 27, 8093 Zürich, Switzerland

[e] *Max-Planck-Institut für Physik, Werner-Heisenberg-Institut,*
D-80805, München, Germany

E-mail: boehm@mathematik.uni-kl.de, mwittman@rhrk.uni-kl.de,
wuzihao@mail.ustc.edu.cn, yingxu@student.ethz.ch, yzhphy@ustc.edu.cn

Abstract: We present an efficient method to shorten the analytic integration-by-parts (IBP) reduction coefficients of multi-loop Feynman integrals. For our approach, we develop an improved version of Leinartas' multivariate partial fraction algorithm, and provide a modern implementation based on the computer algebra system `Singular`. Furthermore, we observe that for an integral basis with uniform transcendental (UT) weights, the denominators of IBP reduction coefficients with respect to the UT basis are either symbol letters or polynomials purely in the spacetime dimension D. With a UT basis, the partial fraction algorithm is more efficient both with respect to its performance and the size reduction. We show that in complicated examples with existence of a UT basis, the IBP reduction coefficients size can be reduced by a factor of as large as ~ 100. We observe that our algorithm also works well for settings without a UT basis.

Keywords: Scattering Amplitudes, Differential and Algebraic Geometry

ArXiv ePrint: 2008.13194

https://doi.org/10.1007/JHEP12(2020)054

Contents

1 Introduction

With the end of Large Hadron Collider (LHC) run-II and the upgrade to HL-LHC [1, 2], there is an eager demand for high-precision physics computations. The integration-by-parts (IBP) reduction is a critical and often bottleneck step for the evaluation of multi-loop scattering amplitudes in precision physics. Via the IBP identities [3, 4], a large number of Feynman integrals can be reduced to a small set of master integrals.

There are many publicly available IBP reduction programs, like AIR, FIRE, Kira, Reduze and LiteRed [5–16], based on the Laporta algorithm [17] and/or the algebra structures of IBP relations [18–20]. In recent years, many new ideas and programs have appeared for use in the computation of complicated multi-loop IBP reductions, for example, syzygy approach [21–26], finite-field interpolation [27–31], module intersection [32, 33], intersection theory [34–37], η expansion [38–42] and direct solution of IBP recursive relations [43].

Besides the development of the computational techniques for IBP reductions, there is another problem which was less addressed in the literature. Frequently, after an analytic

IBP reduction of complicated multi-loop Feynman integrals, we obtain reduction coefficients with a huge size, as rational functions of the spacetime parameter D and kinematic variables. The huge coefficients are difficult to store, to transfer, to use for analytic scattering amplitude computations, and also very cumbersome for numerical evaluations. Thus, an important question arises:

How do we simplify the analytic IBP reduction coefficients in practice?

One natural idea to make analytic IBP reduction coefficients shorter, is to choose a "good" master integral basis. Early attempts were made to test different integral orderings in the Laporta algorithm, in order to get shorter reduction coefficients. However, it is difficult to dramatically shorten IBP reduction coefficients by simply changing the integral ordering. Recently, new methods were presented [44, 45] to find a good master integral basis such that the dimensional parameter D factorizes out in the final IBP reduction coefficients and makes the reduction much easier. In ref. [33], the master integral basis with uniform transcendental (UT) weights [46, 47] was suggested to shorten the size of IBP reduction coefficients.

In this paper, we propose a powerful method to reduce the byte size of the analytic IBP reduction coefficients, which is based on our modern version of Leinartas' multivariate partial fraction algorithm [48, 49]. Leinartas' algorithm has been used for solving basis transformation matrix in Meyer's UT determination algorithm [50], and for the reconstruction and simplification of the planar two-loop five-parton pentagon function coefficients [51]. We develop an improved version of Leinartas' algorithm and implement it in a library for the open source computer algebra system SINGULAR [52]. From the examples we have tested, this method can rewrite a huge rational function in IBP reduction coefficients as a much shorter sum of simpler rational functions.

The improvements to Leinartas' algorithm include an additional decomposition step between the first step (Nullstellensatz decomposition) and second step (algebraic dependence decomposition) of the algorithm which reduces the size of the denominators (and numerators) by doing a (multivariate) division with remainder by the denominator factors. Moreover, in addition to Leinartas' original algorithm, we add a third step in the algorithm, which implements a numerator decomposition as suggested in [50] and uses a syzygy computation to reduce the size of the decomposition expression. In particular in the case of examples arising from IBP reductions, due to the additional decomposition step and by reducing the size of the algebraic relations used, we were able to drastically reduce the runtime of the second step of Leinartas' algorithm, which relies on algebraic relations between the denominator factors. For this we make use of SINGULAR's efficient algorithms for calculating Gröbner bases, syzygy modules and polynomial factorizations. We provide a detailed description of the algorithm in pseudocode.

As an algorithm based on partial fractioning, the size reduction ratio and the running time depend on the degree of irreducible denominators. We combine our partial fractioning approach with the strategy of choosing a "good" master integral basis. In particular, as mentioned in ref. [33], we suggest that when a UT master integral basis for the integral family under consideration exists, it is advantageous to first reduce Feynman integrals to the UT basis, and then run our partial fraction algorithm to shorten the size of the IBP coefficients. The reason is that, in the examples we have tested, for Feynman integrals

$$G[\alpha_1, \ldots, \alpha_j] = \int \prod_{j=1}^{L} \frac{d^D l_j}{i\pi^{D/2}} \frac{1}{\prod_{i=1}^{n} D_i^{\alpha_i}}, \quad \alpha_i \in \mathbb{Z} \tag{1.1}$$

with each D_i defined as a square of a $\mathbb{Z}$-linear combination of loop and external momenta minus the mass term, IBP reduction coefficients with respect to a UT basis have the following good properties:

- The spacetime dimension parameter D factorizes out in the denominator of the reduction coefficients.
- Except the factors purely in D's, the other factors in the IBP reduction coefficients' denominators, are (a subset of) the symbol letters.

Therefore, using a UT basis, we usually get much simpler irreducible factors in the denominators of IBP reduction coefficients. This property makes the partial fractioning much faster and the result usually shorter than that from the usual master integral choice.

We tested various IBP reduction coefficients from simple diagrams to complicated frontier diagrams. In some complicated IBP reduction coefficients examples, we observe that our partial fractioning algorithm, combined with the UT basis choice, dramatically shortens the coefficient size by a factor of as large as 100. In the appendix B, we explicitly list an example of one coefficient, before and after the partial fraction decomposition to provide an impression of this dramatic reduction of size.

We distribute the Singular code of our partial fraction implementation as an open source Singular library for download:

https://github.com/Singular/Singular/tree/spielwiese/Singular/LIB/pfd.lib

This paper is organized as follows: in section 2 we set up the notations and review the concepts of IBP reduction and master integrals. In section 3, we present our improved verion of Leinartas' algorithm to shorten IBP reduction coefficients. In section 4, we provide several IBP reduction simplifications, and also emphasize the benefit of using UT bases in case they exist. In section 5, we summarize our discoveries and discuss possible directions for future research. In the appendices, we provide a manual describing the use of our Singular library for multivariate partial fractioning, and an explicit example of the coefficient size reduction.

2 IBP and master integrals

2.1 Integration-by-parts identities and master integrals

There are many algebraic relations between different Feynman integrals and it is very efficient to use these relations to obtain further Feynman integrals from the ones we already know. A very useful set of relations can be obtained via the integration-by-parts (IBP) identities, which relate different integrals of a given integral family.

Consider a Feynman integral with any loops

$$\int \prod_{j=1}^{L} \frac{d^D l_j}{i\pi^{D/2}} \frac{1}{\prod_{i=1}^{n} D_i^{\alpha_i}}, \tag{2.1}$$

where L is the number of loops, α_i are integer indisces and the denominators are given by

$$D_i = \sum_{j \geq k \geq 1}^{L} A_i^{jk} l_j \cdot l_k + \sum_{i=1}^{L} B_i^j \cdot l_j + E_i, \tag{2.2}$$

i.e are quadratic or linear functions of the external momenta p_i and the loop momenta l_i.

The standard IBP relation [3, 4] is,

$$0 = \int \prod_{j=1}^{L} \frac{d^D l_j}{i\pi^{D/2}} \frac{\partial}{\partial l_m} \left(q_k \prod_{i=1}^{n} D_i^{-\alpha_i} \right), \tag{2.3}$$

where $m = 1, \ldots, L$ with q_k a linear combination of loop momenta and external momenta.

With the IBP identities, we can find the basis of a given integral family, which are called master integrals (MIs). The fact that the number of master integrals is always finite was proven in ref. [53].

So a Feynman integral can be written as a linear combination of master integrals,

$$I[\alpha_1, \ldots, \alpha_n] = \sum_i c_i I_i, \tag{2.4}$$

here α_i are integer indices of denominators and I_i are master integrals.

In practice, IBP reduction can be done by many algorithms, such as the Laporta algorithm [17], the algebra structures of IBP relations [18–20], finite-field interpolation [27–31], module instersection [32], intersection theory [34], η expansion [39] and direct solution of IBP recursive relations [43]. And there are also many public IBP reduction codes, like AIR, FIRE, Kira, Reduze, LiteRed [5–15].

2.2 Differential equation and UT basis

Since the master integrals are functions of scalar products of external momenta, it is natural to consider the derivatives with respect to the scalar products. By introducing a vector

$$\vec{I} = \begin{pmatrix} I_1 \\ I_2 \\ \ldots \\ I_n \end{pmatrix}, \tag{2.5}$$

here I_i are the master integrals of a corresponding Feynman diagram, we can set up the following differential equation

$$d\vec{I} = (dA)\vec{I}, \tag{2.6}$$

where A is a $n \times n$ matrix. If we write down a differential equation using scalar products, then every element of A is a rational function of scalar products and spacetime dimension D.

While, Henn showed that with a new choice of MIs, differential equations can simplify in a way that they can be solved easily order by order [46, 47]. With suitable MIs, the differential equation can be written like that

$$d\vec{I}' = \epsilon(dA)\vec{I}' \tag{2.7}$$

with

$$A = \sum A_k \log S_k. \tag{2.8}$$

This is called the canonical form of differential equations, here we set $D = 4 - 2\epsilon$ and each A_k is a constant matrix, S_k are functions of Lorentz invariants, which are called symbol letters.

With (2.7), the differential equations can be solved order by order in an ϵ-order expansion:

$$\begin{gathered}\vec{I}' = \vec{I}'_0 + \epsilon\vec{I}'_1 + \epsilon^2\vec{I}'_2 + \ldots, \\ d\vec{I}'_1 = (dA)\vec{I}'_0, \quad d\vec{I}'_2 = (dA)\vec{I}'_1 \ldots.\end{gathered} \tag{2.9}$$

The key property of these suitable master integrals can be described with the concept of the degree of transcendentality $\mathcal{T}(f)$ of a function. $\mathcal{T}(f)$ defines the fold number of iterated integrals needed in the function f. Moreover, we require $\mathcal{T}(f_1 f_2) = \mathcal{T}(f_1) + \mathcal{T}(f_2)$. So that, we can see

$$\begin{gathered}\mathcal{T}(\mathrm{Li}_k(x)) = k, \qquad \mathcal{T}(\log x) = 1, \qquad \mathcal{T}(\zeta_n) = \mathcal{T}(\mathrm{Li}_n(1)) = n, \\ \mathcal{T}(\text{algebraic factors}) = 0, \qquad \mathcal{T}(\zeta_2) = \mathcal{T}\left(\frac{\pi^2}{6}\right) = 2 \Rightarrow \mathcal{T}(\pi) = 1.\end{gathered} \tag{2.10}$$

If the function also satisfies

$$\mathcal{T}\left(\frac{d}{dx}f(x)\right) = \mathcal{T}(f(x)) - 1, \tag{2.11}$$

then the function f is called a pure function. With this definition we can see that if we multiply a pure function with an algebraic function of x, the resulting function would still have the same uniform transcendentality but no longer be a pure function anymore, since the derivative is also applied on the algebraic function.

Because of (2.7) and (2.9), we can see that the functions in $\vec{I}'_k$ are all pure functions, hence the $\vec{I}'$ is called uniform transcendental (UT) basis.

There are many ways to construct a UT basis. For examples, we can construct it via Fuchsia and epsilon, based on the Lee's algorithm [54–56]. Meyer proposed a package CANONICA to find a transformation to get UT integrals [50]. What is more, by means of leading singularity analysis and the dlog ansatz, we can also construct a UT basis [57]. A UT basis can be also constructed via Baikov analysis [58], and systematically via the dlog form in a general representation and the intersection theory [59]. And recently, it was discovered that the full UT basis from only one UT integral [60].

2.3 Symbol of a transcendental function

In section 2.2, we proposed the canonical form of differential equation in (2.7) and (2.8). In the case where the symbol letter alphabet can be written in terms of rational functions (in at least one variable), one can write the answer in terms of Goncharov polylogarithms (also called hyperlogarithms, multiple logarithms) [47, 61]. The Goncharov polylogarithms can be defined iteratively as follows,

$$G(a_1,\ldots,a_n;z) = \int_0^z \frac{dt}{t-a_1} G(a_2,\ldots,a_n;t), \quad a_i \in \mathbb{C}, \tag{2.12}$$

with

$$G(z) \equiv G(;z) = 1. \tag{2.13}$$

In the special case where all the a_i are zero, we define, using the obvious vector notation $\vec{a}_n = (a, ..., a)$, $a \in \mathbb{C}$,

$$G(\vec{0}_n;z) = \frac{1}{n!}\log^n z, \quad G(\vec{a}_n;z) = \frac{1}{n!}\log^n\left(1-\frac{z}{a}\right). \tag{2.14}$$

A Goncharov polylogarithm T_k of transcendentality degree k can be written as a linear combination (with rational coefficients) of k-fold iterated integrals of the form [62]

$$T_k = \int_a^b d\log R_1 \circ \ldots \circ d\log R_k, \tag{2.15}$$

where a and b are rational numbers, $R_i(t)$ are rational functions with rational coefficients and the iterated integrals are defined recursively by

$$\int_a^b d\log R_1 \circ \ldots \circ d\log R_k = \int_a^b \left(\int_a^t d\log R_1 \circ \ldots \circ d\log R_{k-1}\right) d\log R_k(t), \tag{2.16}$$

in physics, there $d\log R_k$ are just the ones appeared in eq (2.7), with R_k equal S_k in (2.8).

There is one useful quantity associated with T_k called the symbol, which is an element of the k-fold tensor product of rational functions modulo constants [63], denoted by S. The symbol of the function T_k is

$$\text{symbol}(T_k) \equiv S(T_k) = R_1 \otimes R_2 \otimes \ldots \otimes R_k, \tag{2.17}$$

that is why S_k in (2.8) are called symbol letters.

There are many other properties of the symbol, see refs. [64, 65] for a discussion of their properties.

3 Improved Leinartas' algorithm and modern implementation

In this section we describe an algorithm based on the work of Leinartas [48, 49] to reduce the size of rational functions by writing them as a sum of functions with "smaller" numerators and denominators.

The improvements to the algorithm described in the original paper by Leinartas lie mainly in an additional decomposition step described in algorithm 4, which reduces the size of the numerators and denominators by doing a (multivariate) division with remainder by the denominator factors, as well as the changes discussed in Remark 3.13, which aim at improving the performance of the second decomposition step (algorithm 2) by reducing the size of the required annihilating polynomials. In addition to Leinartas' original algorithm we also add, as suggested in [50], a numerator decomposition as the final step of the algorithm and use a syzygy module computation to reduce the size of the decomposition (see algorithm 3 and Remark 3.11). Thus, while Leinartas' original algorithm calculates a decomposition satisfying only the first two conditions in Theorem 3.5, we add an additional condition. In our implementation, we make use of the computer algebra system Singular, which provides efficient algorithms for the calculation of Gröbner bases and syzygy modules as well as polynomial factorization.

To state more precisely what we mean by "smaller" numerators/denominators, we first need the following definitions. The goal is then an algorithmic proof of Theorem 3.5. For this, let in the following $K[x_1,\dots,x_d]$ or short $K[\mathbf{x}]$ be the polynomial ring over some field K in d variables $\mathbf{x}=(x_1,\dots,x_d)$ and let $\overline{K}$ denote the algebraic closure of K.

Definition 3.1. (algebraic dependence) A set $\{q_1,\dots,q_m\}\subseteq K[\mathbf{x}]$ of m polynomials is called *algebraically dependent* if there exists a nonzero polynomial $p\in K[y_1,\dots,y_m]$ in m variables, such that $p(q_1,\dots,q_m)=0$ in $K[\mathbf{x}]$. Call p an *annihilating polynomial* of $q_1,\dots,q_m$.

Definition 3.2 (monomial ordering)**.** A *monomial ordering* for $K[\mathbf{x}]$ is a total ordering "$>$" on the set $\left\{\mathbf{x}^\alpha\middle|\alpha\in\mathbb{N}^d\right\}$ of monomials (writing "$\mathbf{x}^\alpha$" for $x_1^{\alpha_1}\cdot\ldots\cdot x_d^{\alpha_d}$), such that $>$ is compatible with multiplication, i.e. for all $\alpha,\beta,\gamma\in\mathbb{N}^d$ it holds

$$\mathbf{x}^\alpha>\mathbf{x}^\beta\Rightarrow\mathbf{x}^\alpha\mathbf{x}^\gamma>\mathbf{x}^\beta\mathbf{x}^\gamma \tag{3.1}$$

and $>$ is called *global* if it is a well ordering or equivalently if $1<x_i$ for all $i=1,\dots,d$. For any polynomial $f\in K[\mathbf{x}]$ write $L(f)$ for its *lead monomial*, that is the largest monomial with respect to $>$.

Definition 3.3 (Gröbner basis)**.** A *Gröbner basis* of an ideal $I\subseteq K[\mathbf{x}]$ with respect to a given global monomial ordering is a finite subset $G\subseteq I$ such that the ideals generated by all lead monomials of G and of I coincide:

$$\langle L(g)|g\in G\rangle=\langle L(f)|f\in I\rangle \tag{3.2}$$

Definition 3.4 (division with remainder)**.** After the choice of a (global) monomial ordering there exists an algorithm (multivariate reduced division with remainder, (see [66], section 3 Theorem 3) to determine for any polynomials $f,g_1,\dots,g_r\in K[\mathbf{x}]$ a division expression

$$f=r+\sum_{i=1}^r a_ig_i \qquad (r,a_1,\dots,a_r\in K[\mathbf{x}]) \tag{3.3}$$

such that none of the lead monomials $L(a_i g_i)$ are bigger than $L(f)$ and no term of r is divisible by any lead monomial $L(g_i)$. Call a polynomial r with this property *reduced* with respect to $g_1, \dots, g_r$.

In case $G = (g_1, \dots, g_r)$ is a Gröbner basis of an ideal $I \subseteq K[\mathbf{x}]$, it can be shown ([66], section 6 Proposition 1), that the remainder r only depends on the monomial ordering and I. In this case call r *reduced* with respect to I.

Call G a *reduced Gröbner basis*, if every $g \in G$ is reduced with respect to $G \backslash \{g\}$. It can be shown ([66], section 7 Proposition 6), that for any ideal $I \subseteq K[\mathbf{x}]$ a reduced Gröbner basis exists and is unique up to multiplication with constants and reordering of the elements. It can be calculated with Buchberger's algorithm ([66], section 7 Theorem 2).

Theorem 3.5 (partial fraction decomposition)**.** *Let $f, g \in K[\mathbf{x}]$ and let $g = \prod_{i=1}^m q_i^{e_i}$ be the factorization of g into irreducible factors ($e_i \in \mathbb{N}$). Then there exists a decomposition*

$$\frac{f}{g} = \sum_{S \subseteq \{1,\dots,m\}} \frac{f_S}{\prod_{i \in S} q_i^{b_i}} \qquad (b_i \in \mathbb{N}, f_S \in K[\mathbf{x}]) \tag{3.4}$$

where all nonzero summands satisfy the following conditions

(1) the polynomials $\{q_i | i \in S\}$ have a common zero in $\overline{K}^d$

(2) the polynomials $\{q_i | i \in S\}$ are algebraically independent

(3) f_S is reduced with respect to the ideal $\langle q_i | i \in S \rangle \subseteq K[\mathbf{x}]$

Note that (3) depends on the monomial ordering. In order to get numerator polynomials of low degree, a degree ordering (i.e. $\deg(\mathbf{x}^\alpha) < \deg(\mathbf{x}^\beta) \Rightarrow \mathbf{x}^\alpha < \mathbf{x}^\beta$ for any monomials $\mathbf{x}^\alpha, \mathbf{x}^\beta$ in $K[\mathbf{x}]$) should be chosen. In our Singular implementation we used the graded reverse lexicographic ordering defined by

$$\begin{aligned} \mathbf{x}^\alpha >_{grevlex} \mathbf{x}^\beta :\Leftrightarrow \quad & \deg(\mathbf{x}^\alpha) > \deg(\mathbf{x}^\beta) \text{ or } \deg(\mathbf{x}^\alpha) = \deg(\mathbf{x}^\beta) \\ & \text{and the last nonzero entry of } \alpha - \beta \text{ is negative} \end{aligned} \tag{3.5}$$

Furthermore condition (2) ensures, that at most d different irreducible factors occur in each denominator of the decomposition, since it can be shown, that any set of at least $d+1$ polynomials (in d variables) is algebraically dependent. (This follows directly from the Jacobian criterion 3.7.)

In view of condition (1) in Theorem 3.5, the following corollary to Hilbert's weak Nullstellensatz can be used to eliminate factors from the denominators if the q_i have no common zero.

Lemma 3.6 (Nullstellensatz certificate)**.** *Polynomials $f_1, \dots, f_m \in K[\mathbf{x}]$ have no common zero in $\overline{K}^d$ if and only if the generated ideal is trivial, i.e. $\langle f_1, \dots, f_m \rangle = \langle 1 \rangle = K[\mathbf{x}]$. In this case there exist polynomials $h_1, \dots, h_m \in K[\mathbf{x}]$ such that*

$$1 = \sum_{i=1}^m h_i f_i \tag{3.6}$$

Call $(h_1, \dots, h_m)$ a Nullstellensatz certificate *for $(f_1, \dots, f_m)$.*

Proof. This is exactly the weak Nullstellensatz ([66], secion 4.1 Theorem 1) with the exception, that we require $h_i \in K[\mathbf{x}]$ instead of $h_i \in \overline{K}[\mathbf{x}]$. However the equation $1 = \sum_{i=1}^{m} h_i f_i$ can be seen as a set of linear equations (with coefficients in K) in the coefficients of the polynomials h_i and by the weak Nullstellensatz we know, that it is solvable over $\overline{K}$. But then it is solvable over K as well, since all the coefficients in these linear equations lie in K. Hence we may assume $h_1, \ldots, h_m \in K[\mathbf{x}]$. □

Given a rational function f/g as in Theorem 3.5 for which the irreducible factors q_i of g have no common zero in $\overline{K}^d$, we know that $q_1^{e_1}, \ldots, q_m^{e_m}$ have no common zero as well and if $(h_1, \ldots, h_m)$ is a Nullstellensatz certificate, we can simply multiply f by $1 = \sum_{k=1}^{m} h_k q_k^{e_k}$ to get a decomposition

$$\frac{f}{g} = \frac{f \cdot \sum_{k=1}^{m} h_k q_k^{e_k}}{\prod_{i=1}^{m} q_i^{e_i}} = \sum_{k=1}^{m} \frac{f \cdot h_k}{\prod_{i=1, i\neq k}^{m} q_i^{e_i}} \tag{3.7}$$

where each denominator contains only $m-1$ different irreducible factors.

To calculate this decomposition (algorithm 1), we compute a reduced Gröbner basis G of $\langle q_1^{e_1}, \ldots, q_m^{e_m} \rangle$ as well as the transformation matrix T from the original ideal generators $q_1^{e_1}, \ldots, q_m^{e_m}$ to G. This can be done with Buchberger's algorithm for the computation of Gröbner bases as implemented in the Singular function `liftstd`.

Input: rational function f/g where $f, g \in K[\mathbf{x}]$ and $g = \prod_{i=1}^{m} q_i^{e_i}$ for irreducible $q_i \in K[\mathbf{x}]$
Output: set of rational functions with sum f/g

```
calculate the reduced Gröbner basis $G$ of $\langle q_1^{e_1}, \ldots, q_m^{e_m} \rangle$ as well as the
transformation matrix $T$ from the generators $q_i^{e_i}$ to $G$
if $G = \{c\}$ for $c \in K$ (so $\langle q_1^{e_1}, \ldots, q_m^{e_m} \rangle = \langle 1 \rangle$) then
    using $T$, find polynomials $h_i$ such that $\sum_{i=1}^{m} h_i q_i^{e_i} = 1$ (namely $h_i = T_{i1}/c$)
    return $\left\{ \frac{f \cdot h_k}{\prod_{i=1, i\neq k}^{m} q_i^{e_i}} \middle| k = 1, \ldots, m \right\}$
else
    return $\left\{ \frac{f}{g} \right\}$
end if
```

Algorithm 1. `NSSdecompStep` (Nullstellensatz decomposition step).

Note that $\langle q_1^{e_1}, \ldots, q_m^{e_m} \rangle = \langle 1 \rangle$ if and only if $G = \{c\}$ for $c \in K$ (constant polynomial). This follows directly from the uniqueness of reduced Gröbner bases.

Repeated application of algorithm 1 will yield a decomposition satisfying condition (1) in Theorem 3.5. For (2) let's assume f/g is a rational function and $g = \prod_{i=1}^{m} q_i^{e_i}$ as in Theorem 3.5. There is a simple criterion to test for algebraic dependence:

Lemma 3.7 (Jacobian criterion). *A set of m polynomials $\{f_1, \ldots, f_m\} \subseteq K[\mathbf{x}]$ is algebraically independent if and only if the Jacobian matrix $\left(\frac{\partial f_i}{\partial x_j}\right)_{i,j} \in K[\mathbf{x}]^{m \times d}$ has rank m over the field $K(\mathbf{x})$ of rational functions. (A proof can be found in [67].)*

Corollary 3.8. *A set of polynomials $\{q_1, \dots, q_m\} \subset K[\mathbf{x}]$ is algebraically dependent if and only if $\{q_1^{e_1}, \dots, q_m^{e_m}\}$ is $(e_i \in \mathbb{N}_{>0})$.*

Proof. This follows directly from Lemma 3.7 since

$$\left(\frac{\partial\left(q_i^{e_i}\right)}{\partial x_j}\right)_{i,j} = \left(e_i q_i^{e_i-1}\frac{\partial q_i}{\partial x_j}\right)_{i,j} = T \cdot \left(\frac{\partial q_i}{\partial x_j}\right)_{i,j} \tag{3.8}$$

where T is the invertible diagonal matrix $\operatorname{diag}(e_1 q_1^{e_1-1}, \dots, e_m q_m^{e_m-1}) \in K(\mathbf{x})^{m\times m}$. So the Jacobian matrices of $\{q_1, \dots, q_m\}$ and $\{q_1^{e_1}, \dots, q_m^{e_m}\}$ have the same rank over $K(\mathbf{x})$. □

If now the factors $q_1, \dots, q_m$ of the denominator g are algebraically dependent, then so are $q_1^{e_1}, \dots, q_m^{e_m}$ and if $p \in K[\mathbf{y}] = K[y_1, \dots, y_m]$ is an annihilating polynomial we can write

$$p = c_\alpha \mathbf{y}^\alpha + \sum_{\substack{\beta\in\mathbb{N}^m \\ \deg(p)\geq|\beta|\geq|\alpha|}} c_\beta \mathbf{y}^\beta \qquad (c_\alpha, c_\beta \in K, c_\alpha \neq 0) \tag{3.9}$$

such that $c_\alpha \mathbf{y}^\alpha$ is one of the terms of smallest degree (using multi-indices $\beta \in \mathbb{N}^m$, so $\deg(\mathbf{y}^\beta) = |\beta|= \beta_1 + \dots + \beta_m$). Writing $\mathbf{q}$ for the vector $(q_1^{e_1}, \dots, q_m^{e_m})$, it holds

$$\begin{aligned} 0 = p(\mathbf{q}) \quad &\Leftrightarrow \quad c_\alpha \mathbf{q}^\alpha = -\sum_\beta c_\beta \mathbf{q}^\beta \\ &\Leftrightarrow \quad 1 = -\sum_\beta \frac{c_\beta \mathbf{q}^\beta}{c_\alpha \mathbf{q}^\alpha} = -\sum_\beta \frac{c_\beta}{c_\alpha}\prod_{i=1}^m \frac{q_i^{e_i\beta_i}}{q_i^{e_i\alpha_i}} \\ &\Rightarrow \quad \frac{f}{g} = -\sum_\beta \frac{c_\beta}{c_\alpha} f \prod_{i=1}^m \frac{q_i^{e_i\beta_i}}{q_i^{e_i(\alpha_i+1)}} \end{aligned} \tag{3.10}$$

Since $\mathbf{y}^\alpha$ has minimal degree, for every β occurring in equation (3.10) it holds $\beta_i \geq \alpha_i + 1$ for at least one index i. Therefore the factor q_i does not appear in the denominator of the corresponding term. So we obtain a sum of rational functions with at most $m - 1$ different irreducible factors in their denominators and thus, as with algorithm 1, repeated application of this step leads to a decomposition satisfying condition (2) in Theorem 3.5. But in order to turn this into an algorithm, we need a way of computing annihilating polynomials:

Lemma 3.9 (annihilating polynomials)**.** *The annihilating polynomials of a fixed tuple $(f_1, \dots, f_m)$ of polynomials in $K[\mathbf{x}] = K[x_1, \dots, x_d]$ are precisely the elements of the ideal*

$$\langle y_1 - f_1, \dots, y_m - f_m\rangle_{K[\mathbf{x},\mathbf{y}]} \cap K[\mathbf{y}] \quad \textit{where} \quad \mathbf{y} = (y_1, \dots, y_m).$$

Proof. Let $p \in K[\mathbf{y}]$ be an annihilating polynomial for $f_1, \dots, f_m$ and write $\mathbf{f} = (f_1, \dots, f_m)$. Define $\tilde{p}(\mathbf{x}, \mathbf{y}) = p(\mathbf{f} - \mathbf{y}) \in K[\mathbf{x}, \mathbf{y}]$. Then $\tilde{p}(\mathbf{x}, \mathbf{0}) = p(\mathbf{f}) = 0$ and thus every term of $\tilde{p}$ must be divisible by some y_i $(1 \leq i \leq m)$, hence $\tilde{p} \in \langle y_1, \dots, y_m\rangle_{K[\mathbf{x},\mathbf{y}]}$. But since $\tilde{p}(\mathbf{x}, \mathbf{f} - \mathbf{y}) = p(\mathbf{y}) = p$ (replacing each y_i by $f_i - y_i$), we get $p \in \langle f_1 - y_1, \dots, f_m - y_m\rangle_{K[\mathbf{x},\mathbf{y}]}$.

Now assume $p \in \langle y_i - f_1, \dots, y_m - f_m\rangle_{K[\mathbf{x},\mathbf{y}]} \cap K[\mathbf{y}]$. Then $p = \sum_{i=1}^m a_i \cdot (y_i - f_i)$ for some $a_i \in K[\mathbf{x}, \mathbf{y}]$ and thus $p(\mathbf{f}) = \sum_{i=1}^m a_i(\mathbf{x}, \mathbf{f}) \cdot (f_i - f_i) = 0$. □

Lemma 3.10 (elimination ordering)**.** *Call a monomial ordering for the polynomial ring $K[\mathbf{x},\mathbf{y}] = K[x_1,\dots,x_d,y_1,\dots,y_m]$ an* elimination ordering *for the variables $x_1,\dots,x_d$ if $L(p) \in K[\mathbf{y}]$ implies already $p \in K[\mathbf{y}]$ for any polynomial $p \in K[\mathbf{x},\mathbf{y}]$.*

If now G is a (reduced) Gröbner basis of an ideal $I \subseteq K[\mathbf{x},\mathbf{y}]$ with respect to such an ordering, then $G \cap K[\mathbf{y}]$ is a (reduced) Gröbner basis of the ideal $I \cap K[\mathbf{y}] \subseteq K[\mathbf{y}]$.

(A proof can be found in [68], Lemma 1.8.3.)

```
Input: rational function $f/g$ where $f, g \in K[\mathbf{x}]$ and $g = \prod_{i=1}^m q_i^{e_i}$ for irreducible
$q_i \in K[\mathbf{x}]$
Output: set of rational functions with sum $f/g$

1: if rank $\left(\frac{\partial q_i}{\partial x_j}\right)_{i\le m, j\le d} < m$ then
2:     calculate the reduced Gröbner basis $G$ of $\langle y_1 - q_1^{e_1}, \dots, y_m - q_m^{e_m}\rangle \subseteq K[\mathbf{x},\mathbf{y}]$
       with respect to an elimination ordering for $x_1,\dots,x_d$ ($\mathbf{y} = (y_1,\dots,y_m)$)
3:     $G' = G \cap K[\mathbf{y}]$
4:     $p =$ some element of $G'$ (choose a "simple" one, e.g. with smallest degree)
5:     write $p = c_\alpha y^\alpha + \sum_\beta c_\beta y^\beta$ where $y^\alpha$ has minimal degree
6:     return $\left\{-\frac{c_\beta}{c_\alpha} f \prod_{i=1,i\neq k}^m q_i^{e_i\cdot(\beta_i-\alpha_i-1)} \middle| \beta \in \mathbb{N}^m\right\}$
7: else
8:     return $\left\{\frac{f}{g}\right\}$
9: end if
```

Algorithm 2. `algDependDecompStep` (algebraic dependence decomposition step).

In order to calculate the rank of the Jacobian matrix in line 1 of algorithm 2, we can test, if the syzygy module of the $K[\mathbf{x}]$-module generated by the rows of the Jacobian matrix (that is the module of all $K[\mathbf{x}]$-linear relations of the rows) is zero (e.g. with the Singular command `syz`). Instead of calculating the rank of the Jacobian, we could also just check whether G' is empty, however the derivatives $\frac{\partial q_i}{\partial x_j}$ are in general of much lower degree than $q_i^{e_i}$, so using the Jacobian criterion is cheaper, especially for small factors q_i. Also, if $d < m$ the criterion becomes trivial since the rank is at most d.

The previous two strategies to decompose a rational function only decrease the size of the denominators while leaving the numerator mostly untouched. To simplify the numerators as well, it makes sense to do a (reduced) division with remainder of the numerator f by a Gröbner basis G of the ideal $\langle q_1,\dots,q_m\rangle$ generate by all the irreducible factors in the denominator. This gives a division expression

$$f = r + \sum_{g\in G} a_g g \qquad (r, a_g \in K[\mathbf{x}]) \tag{3.11}$$

as in Definition 3.4. Rewriting this in terms of the ideal generators $q_1,\dots,q_m$ we get

$$\begin{aligned} f &= r + \sum_{k=1}^m b_k q_k \qquad (b_k \in K[\mathbf{x}]) \\ \Rightarrow \quad \frac{f}{g} &= \frac{r}{\prod_{i=1}^m q_i^{e_i}} + \sum_{k=1}^m \frac{b_k}{q_k^{(e_k-1)} \prod_{i=1,i\neq k}^m q_i^{e_i}} \end{aligned} \tag{3.12}$$

The first term with numerator r already fulfills condition (3) of Theorem 3.5 and all other terms have one irreducible factor q_i less in their denominator. Thus repeated application of algorithm 3 results in a decomposition satisfying (3).

```
Input: rational function $f/g$ where $f, g \in K[\mathbf{x}]$ and $g = \prod_{i=1}^{m} q_i^{e_i}$ for irreducible $q_i \in K[\mathbf{x}]$
Output: set of rational functions with sum $f/g$

1: calculate the reduced Gröbner basis $G$ of $\langle q_1, \dots, q_m \rangle$ as well as the
   transformation matrix $T$ from the generators $q_i$ to $G$
2: divide $f$ by $G$ (reduced division with remainder) to get a division expression
   $f = r + \sum_{g \in G} a_g g$ where $r, a_g \in K[x_1, \dots, x_d]$ and $r$ is reduced w.r.t. $G$
3: using $T$ and $(a_g)_{g \in G}$, find polynomials $b_k$ such that $f = r + \sum_{k=1}^{m} b_k q_k$
4: replace $(b_1, \dots, b_m) \in K[\mathbf{x}]^m$ by its remainder after division by a Gröbner basis of
   the syzygy module of $(q_1, \dots, q_m)$
5: return $\left\{ \frac{b_k}{q_k^{(e_k-1)} \prod_{i=1, i \neq k}^{m} q_i^{e_i}} \middle| k = 1, \dots, m,\ b_k \neq 0 \right\} \cup \left\{ \frac{r}{g} \right\}$
```

Algorithm 3. `numeratorDecompStep` (numerator decomposition step).

Remark 3.11 (syzygy reduction)**.** In order to further simplify the coefficients b_k, in line 4 of algorithm 3 we replace $(b_1, \dots, b_m) \in K[\mathbf{x}]^m$ by its remainder after division by (a Gröbner basis of) the syzygy module $\{(s_1, \dots, s_m) | \sum_{i=1}^{m} s_i q_i = 0\} \subset K[\mathbf{x}]^m$ of $(q_1, \dots, q_m)$ (extending the notions of monomial orderings, Gröbner bases and division with remainder from $K[\mathbf{x}]$ to the $K[\mathbf{x}]$-module $K[\mathbf{x}]^m$ as described in [68], Chapter 2.3). After doing this, the polynomials b_k will still satisfy equation (3.12), since we just changed $(b_1, \dots, b_m)$ by an element $(s_1, \dots, s_m)$ of the syzygy module and $\sum_{i=1}^{m} s_i q_i = 0$. This step is optional, but in practice we found, that reducing the coefficients by the syzygy module can dramatically reduce the runtime of algorithm 5. In SINGULAR we can just apply the procedures `syz` and `std` to calculate a Gröbner basis of the syzygy module and reduce the coefficients b_k with `reduce`.

Using all three decomposition techniques one after the other yields algorithm 5 which calculates a partial fraction decomposition fulfilling conditions (1), (2) and (3), finally proving Theorem 3.5.

However, in practice the calculation of annihilating polynomials can be quite slow if the degrees of the polynomials $q_i^{e_i}$ get too big. Therefore it is more efficient to do an additional "short" numerator decomposition before the algebraic dependence decomposition (see algorithm 5), in order to simplify the denominators. For this we repeatedly apply algorithm 4, which is identical to algorithm 3 with the exception, that whenever the remainder r is nonzero, we return the input and do not decompose further since the term corresponding to r would not have a smaller denominator anyway. Note that this is only effective, because most of the rational functions we are interested in (i.e. those arising from IBP-reductions) have the property, that the numerator is already contained in the ideal

$\langle q_1, \dots, q_m \rangle$, such that the remainder r becomes 0. Thus, while it is not needed in order to get a decomposition fulfilling the conditions of Theorem 3.5, the insertion of a short numerator decomposition (lines 6 to 8 in algorithm 5) reduces runtimes.

Input: rational function f/g where $f, g \in K[\mathbf{x}]$ and $g = \prod_{i=1}^{m} q_i^{e_i}$ for irreducible $q_i \in K[\mathbf{x}]$
Output: set of rational functions with sum f/g

```
calculate the reduced Gröbner basis $G$ of $\langle q_1, \dots, q_m \rangle$ as well as the
  transformation matrix $T$ from the generators $q_i$ to $G$
divide $f$ by $G$ (reduced division with remainder) to get a division expression
  $f = r + \sum_{g \in G} a_g g$ where $r, a_g \in K[x_1, \dots, x_d]$ and $r$ is reduced w.r.t. $G$
if $r \neq 0$ then
    return $\left\{ \frac{f}{g} \right\}$
end if
using $T$ and $(a_g)_{g \in G}$, find polynomials $b_k$ such that $f = r + \sum_{k=1}^{m} b_k q_k$
replace $(b_1, \dots, b_m) \in K[\mathbf{x}]^m$ by its remainder after division by a Gröbner basis
  of the syzygy module of $(q_1, \dots, q_m)$
return $\left\{ \frac{b_k}{q_k^{(e_k - 1)} \prod_{i=1, i \neq k}^{m} q_i^{e_i}} \,\middle|\, k = 1, \dots, m,\ b_k \neq 0 \right\}$
```

Algorithm 4. `shortNumeratorDecompStep` (short numerator decomposition step).

Lemma 3.12. *algorithm 5 terminates for any input f/g and returns a partial fraction decomposition of f/g satisfying all three conditions in Theorem 3.5.*

Proof. As shown above, Algorithms 1 to 4 applied to any rational function f/g always return a set of rational functions with sum f/g. So in algorithm 5, at all times the elements of D sum up to the input of the algorithm.

It is also easy to see, that if algorithm 3.5 terminates, the returned decomposition D indeed fulfils conditions (1) to (3): if no term of the decomposition is decomposed further when applying algorithm 1 (i.e. the first `while` loop terminates), then by Lemma 3.6 in each denominator the irreducible factors q_i have a common zero. Similarly, if a rational function is not decomposable by algorithm 2, then by Lemmata 3.9 and 3.10 and Corollary 3.8 the q_i are algebraically independent. And finally, if a rational function f/g is not decomposed further in algorithm 3, this means that $f = r$ in equation (3.12) and thus the numerator is reduced with respect to the ideal generated by the factors q_i in the denominator.

Also, algorithm 2, 3 and 4 only ever change the denominators by removing irreducible factors (and changing their exponents) and thus preserve the properties (1) and (2) in Theorem 3.5. (If $q_1, \dots, q_m$ have a common zero or are algebraically independent, then so are $q_1, \dots, q_{m-1}$.)

It remains to show that all `while` loops terminate. As argued above, each term in the decomposition returned by algorithm 1 has fewer different irreducible factors in its denominator than the input and thus after applying `NSSdecompStep` to each element of D,

Input: rational function f/g where $f, g \in K[x_1, \ldots, x_d]$
Output: partial fraction decomposition as a set of rational functions.

```
factorize $g = \prod_{i=1}^m q_i^{e_i}$ where $q_i \in K[x_1, \ldots, x_d]$ are irreducible
(and represent all denominators in the following steps in factorized form)
$D = \{f/g\}$
while $\exists s \in D$ such that $|\texttt{NSSdecompStep}(s)| > 1$ do
    $D = \texttt{NSSdecompStep}(s) \cup D \backslash \{s\}$
    and merge elements of $D$ with equal denominators
end while
while $\exists s \in D$ such that $|\texttt{shortNumeratorDecompStep}(s)| > 1$ do
    $D = \texttt{shortNumeratorDecompStep}(s) \cup D \backslash \{s\}$
    and merge elements of $D$ with equal denominators
end while
while $\exists s \in D$ such that $|\texttt{algDependDecompStep}(s)| > 1$ do
    $D = \texttt{algDependDecompStep}(s) \cup D \backslash \{s\}$
    and merge elements of $D$ with equal denominators
end while
while $\exists s \in D$ such that $|\texttt{numeratorDecompStep}(s)| > 1$ do
    $D = \texttt{numeratorDecompStep}(s) \cup D \backslash \{s\}$
    and merge elements of $D$ with equal denominators
end while
return $D$
```

Algorithm 5. Partial fraction decomposition.

all terms have at most $m - 1$ different irreducible factors q_i and thus the first loop terminates by induction on m. The same argument also works for `algDependDecompStep`. In `numeratorDecompStep` each element of the returned decomposition has one less irreducible factor in its denominator than the input with the exception of the term corresponding to the remainder r in equation (3.12), which has the same denominator as the input. However terms of this form are not decomposed further (since in equation (3.12) r is already reduced with respect to $\langle q_1, \ldots, q_m \rangle$) and can thus be disregarded in the argument. Now by induction on $\sum_{i=1}^m e_i$ the fourth loop terminates as well. A very similar argument works for `shortNumeratorDecompStep`. □

Since the calculation of annihilating polynomials can still be quite slow for some of the more complicated rational functions, we make the following modification to the algorithm.

Remark 3.13 (simplified `algDependDecompStep`)**.** In algorithm 2 it is also possible to use an annihilating polynomial for $q_1, \ldots, q_m$ rather than $q_1^{e_1}, \ldots, q_m^{e_m}$. Instead of equation (3.10) we then get the decomposition

$$\frac{f}{g} = -\sum_\beta \frac{c_\beta}{c_\alpha} f \prod_{i=1}^m \frac{q_i^{\beta_i}}{q_i^{\alpha_i + e_i}}. \tag{3.13}$$

where $p = c_\alpha \mathbf{y}^\alpha + \sum_\beta c_\beta \mathbf{y}^\beta$ is the annihilating polynomial and $c_\alpha y^\alpha$ a term of minimal degree as in equation (3.9). Since the polynomials q_i are of lower degree than $q_i^{e_i}$, this will speed up the calculation of annihilating polynomials at the cost of needing more steps in the algebraic dependence decomposition in algorithm 5, since the number of different irreducible denominator factors then does not decrease in every step. (If $\beta_i < \alpha_i + e_i$ for all i, it stays the same.) In fact it is not at all clear that algorithm 5 terminates with the simplified `algDependDecompStep` and indeed this depends on the choice of α. However, if α is chosen minimal with respect to the graded reverse lexicographic ordering $>_{grevlex}$ on $K[\mathbf{y}]$ as defined in equation (3.5), it can be shown, that algorithm 5 still terminates with a correct decomposition:

Proof. All we have to show is that the third `while` loop in algorithm 5 terminates, the rest follows as in the proof of Lemma 3.12. For this, take any sequence $f_1/g_1, f_2/g_2, \ldots$ of rational functions such that f_{i+1}/g_{i+1} is one of the terms in `algDependDecompStep`(f_i/g_i). It is enough to show, that in each such sequence eventually a rational function is reached, which has fewer *different* factors in its denominator or satisfies (2) already. Assume this is not the case. Then all the denominators g_i have the same irreducible factors (with different exponents). Thus in each call of the simplified `algDependDecompStep` the same annihilating polynomial p is chosen (assuming a deterministic implementation of the algorithm). Write

$$p = c_\alpha \mathbf{y}^\alpha + \sum_{j=1}^{r+s} c_{\beta^{(j)}} \mathbf{y}^{\beta^{(j)}} \qquad (r, s \in \mathbb{N}) \tag{3.14}$$

where $|\beta^{(j)}| = |\alpha|$ for $j \leq r$ and $|\beta^{(j)}| > |\alpha|$ for $j > r$. Since $\mathbf{y}^\alpha$ is minimal with respect to $>_{grevlex}$, for each $j = 1, \ldots, r$ there exists an index $k_j \in \{1, \ldots, m\}$ such that

$$\alpha_{k_j} > \beta^{(j)}_{k_j} \text{ and } \alpha_i = \beta^{(j)}_i \quad \text{for all } i > k_j. \tag{3.15}$$

Without loss of generality we may assume that $k_1 \geq \ldots \geq k_r$. If we factorize the denominators as $g_i = \prod_{l=1}^m q_l^{e_l^{(i)}}$ $(e_l^{(i)} \in \mathbb{N})$, then it holds

$$\frac{f_{i+1}}{g_{i+1}} = -\frac{c_{\beta^{(j_i)}}}{c_\alpha} f_i \prod_{l=1}^m \frac{q_l^{\beta_l^{(j_i)}}}{q_l^{\alpha_l + e_l^{(i)}}} \tag{3.16}$$

for some index j_i depending on i. Since we assumed that in g_{i+1} no irreducible factor vanishes, it holds

$$e_l^{(i+1)} = \alpha_l + e_l^{(i)} - \beta_l^{(j_i)} > 0 \qquad (l = 1, \ldots, m) \tag{3.17}$$

If $j_i > r$, then the sum of all exponents decreases: $\sum_{l=1}^m e_l^{(i+1)} < \sum_{l=1}^m e_l^{(i)}$. If $j_i \leq r$, it stays the same. Thus the case $j_i > r$ can only occur for finitely many i and after that it always holds $j_i \leq r$, but then $|\beta^{(j_i)}| = |\alpha|$ and by equations (3.15) and (3.17) it holds

$$\sum_{l=1}^{k_{j_i}-1} e_l^{(i+1)} < \sum_{l=1}^{k_{j_i}-1} e_l^{(i)} \quad \text{and} \quad \sum_{l=1}^{k} e_l^{(i+1)} = \sum_{l=1}^{k} e_l^{(i)} \quad \text{for any } k \geq k_{j_i} \tag{3.18}$$

Since the exponents $e_l^{(i)}$ have to stay positive and $k_1 \geq \ldots \geq k_r$, in the sequence $\beta^{(j_1)}, \beta^{(j_2)}, \ldots$ the multi-index $\beta^{(1)}$ can only appear finitely often and after that $\beta^{(2)}$ can only appear finitely often and so on. But $f_1/g_1, f_2/g_2, \ldots$ was an infinite sequence, a contradiction.

Thus in the sequence $f_1/g_1, f_2/g_2, \ldots$ the number of different irreducible factors in the denominators decreases until a rational function is reached, that satisfies condition (2) and the third `while` loop terminates after finitely many iterations. □

Remark 3.14. The multivariate partial fractioning can be combined with the rational reconstruction scheme for commutative algebra developed in [69, 70] as long as a consistent factorization patterns can be guaranteed.

Example 3.15. Consider the rational function

$$\frac{-2s_{12}s_{23}\varepsilon_5 + 3s_{23}s_{34}\varepsilon_5 + s_{15}s_{45}\varepsilon_5 + s_{23}s_{45}\varepsilon_5 - s_{34}s_{45}\varepsilon_5}{8s_{12}s_{23}(-s_{15}+s_{23}+s_{34})(s_{12}-s_{45})s_{45}} \tag{3.19}$$

This is the $(24, 7)$-th entry of the IBP-matrix for the double pentagon (see section 4).[1] For better readability, replace $s_{12}, s_{15}, s_{23}, s_{34}, s_{45}$ by x_1, x_2, x_3, x_4, x_5 and divide by $\varepsilon_5/8$ to get the function

$$\frac{f}{g} = \frac{-2x_1x_3 + 3x_3x_4 + x_2x_5 + x_3x_5 - x_4x_5}{x_1x_3(-x_2+x_3+x_4)(x_1-x_5)x_5} \in \mathbb{R}(\mathbf{x}) = \mathbb{R}(x_1, x_2, x_3, x_4, x_5) \tag{3.20}$$

Now apply algorithm 5 using the graded reverse lexicographic ordering with $x_1 > \cdots > x_5$ and employ the modification to `algDependDecompStep` discussed in Remark 3.13. For simplicity we omit the syzygy reduction step described in Remark 3.11.

1. **factorization of the denominator** The denominator factors as $g = q_1 \cdot q_2 \cdot q_3 \cdot q_4 \cdot q_5$ where $q_1 = x_1$, $q_2 = x_3$, $q_3 = -x_2 + x_3 + x_4$, $q_4 = x_1 - x_5$, $q_5 = x_5$.

2. **Nullstellensatz decomposition** As with most entries of this IBP-matrix, the denominators have already a common zero, namely 0, since none of the factors q_i have a constant term. Thus algorithm 1 (`NSSdecompStep`) does nothing.

3. **short numerator decomposition**

   reduced G.B. of $\langle q_1, q_2, q_3, q_4, q_5 \rangle$: $G = (x_5, x_3, x_2 - x_4, x_1) = (q_5, q_2, q_2 - q_3, q_4 + q_5)$

   reduced division with remainder: $f = (x_2 + x_3 - x_4) \cdot x_5 + (-2x_1 + 3x_4) \cdot x_3$
   $= (x_2 + x_3 - x_4) \cdot q_5 + (-2x_1 + 3x_4) \cdot q_2$

$$\Rightarrow \quad \frac{f}{g} = \frac{x_2 + x_3 - x_4}{q_1 \cdot q_2 \cdot q_3 \cdot q_4} + \frac{-2x_1 + 3x_4}{q_1 \cdot q_3 \cdot q_4 \cdot q_5} = \frac{f_1}{g_1} + \frac{f_2}{g_2} \tag{3.21}$$

   reduced G.B. of $\langle q_1, q_2, q_3, q_4 \rangle$: $G = (x_5, x_3, x_2 - x_4, x_1) = (q_1 - q_4, q_2, q_2 - q_3, q_1)$

   reduced division with remainder: $f_1 = 1 \cdot x_3 + 1 \cdot (x_2 - x_4) = 2 \cdot q_2 + (-1) \cdot q_3$

$$\Rightarrow \quad \frac{f_1}{g_1} = \frac{2}{q_1 \cdot q_3 \cdot q_4} + \frac{-1}{q_1 \cdot q_2 \cdot q_4} \tag{3.22}$$

[1] $\varepsilon_5 \equiv 4i\epsilon_{\mu\nu\rho\sigma}k_1^\mu k_2^\nu k_3^\rho k_4^\sigma$ for the five-point kinematics. This is a square root of a polynomial in Mandelstam variables s_{ij}'s. Here we can treat it as an irrelevant overall factor.

reduced G.B. of $\langle q_1, q_3, q_4, q_5\rangle$: $G = (x_5, x_2 - x_3 - x_4, x_1) = (q_5, -q_3, q_4 + q_5)$

reduced division with remainder: $f_2 = (-2)\cdot x_1 + 3x_4 = (-2)\cdot q_4 + (-2)\cdot q_5 + 3x_4$

The remainder $3x_4$ is nonzero, so we do not decompose f_2/g_2 further and get in total

$$\frac{f}{g} = \frac{2}{q_1\cdot q_3\cdot q_4} + \frac{-1}{q_1\cdot q_2\cdot q_4} + \frac{-2x_1 + 3x_4}{q_1\cdot q_3\cdot q_4\cdot q_5} \tag{3.23}$$

4. **(simplified) algebraic dependence decomposition** For the first two terms the Jacobian matrices have full rank and therefore algorithm 2 (`algDependDecompStep`) does nothing. For the third term the Jacobian matrix is

$$\left(\frac{\partial q_i}{\partial x_j}\right)_{\substack{i=1,3,4,5\\ j=1,2,3,4,5}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0\\ 0 & -1 & 1 & 1 & 0\\ 1 & 0 & 0 & 0 & -1\\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.24}$$

and has only rank 3, so q_1, q_3, q_4, q_5 are algebraically dependent. Indeed it is obvious, that $q_1 = x_1$, $q_4 = x_1 - x_5$ and $q_5 = x_5$ are even linearly dependent and thus a possible annihilating polynomial for q_1, q_3, q_4, q_5 is

$$p = y_1 - y_3 - y_4 \in \mathbb{R}[y_1, y_2, y_3, y_4] \tag{3.25}$$

leading to the relation

$$1 = \frac{q_1 - q_4}{q_5} \quad \Rightarrow \quad \frac{-2x_1 + 3x_4}{q_1\cdot q_3\cdot q_4\cdot q_5} = \frac{-2x_1 + 3x_4}{q_3\cdot q_4\cdot q_5^2} + \frac{2x_1 - 3x_4}{q_1\cdot q_3\cdot q_5^2} \tag{3.26}$$

Now $\{q_3, q_4, q_5\}$ as well as $\{q_1, q_4, q_5\}$ are algebraically independent and we are done. Note that the exponent of q_5 increased and the number of irreducible factors in the denominators stayed the same (4), but the number of *different* irreducible factors decreased from 4 to 3. Overall, we now have

$$\frac{f}{g} = \frac{2}{q_1\cdot q_3\cdot q_4} + \frac{-1}{q_1\cdot q_2\cdot q_4} + \frac{-2x_1 + 3x_4}{q_3\cdot q_4\cdot q_5^2} + \frac{2x_1 - 3x_4}{q_1\cdot q_3\cdot q_5^2} \tag{3.27}$$

and all terms fulfil conditions (1) and (2) of Theorem 3.5.

5. **numerator decomposition** The first two numerators (2 and -1) are obviously reduced with respect to $\langle q_1, q_3, q_4\rangle$ and $\langle q_1, q_2, q_4\rangle$ respectively. So algorithm 3 (`numeratorDecompStep`) does nothing. For the third and fourth term we get:

reduced Gröbner basis of $\langle q_3, q_4, q_5\rangle$: $G = (x_5, x_2 - x_3 - x_4, x_1) = (q_5, -q_3, q_4 + q_5)$

reduced division with remainder: $-2x_1 + 3x_4 = (-2)\cdot x_1 + 3x_4$
$= (-2)\cdot q_4 + (-2)\cdot q_5 + 3x_4$

$$\Rightarrow \quad \frac{-2x_1 + 3x_4}{q_3\cdot q_4\cdot q_5^2} = \frac{-2}{q_3\cdot q_5^2} + \frac{-2}{q_3\cdot q_4\cdot q_5} + \frac{3x_4}{q_3\cdot q_4\cdot q_5^2} \tag{3.28}$$

reduced Gröbner basis of $\langle q_1, q_3, q_5\rangle$: $G = (x_5, x_2 - x_3 - x_4, x_1) = (q_5, -q_3, q_1)$

reduced division with remainder: $2x_1 - 3x_4 = 2 \cdot x_1 - 3x_4 = 2 \cdot q_1 - 3x_4$

$$\Rightarrow \quad \frac{2x_1 - 3x_4}{q_1 \cdot q_3 \cdot q_5^2} = \frac{2}{q_3 \cdot q_5^2} + \frac{-3x_4}{q_1 \cdot q_3 \cdot q_5^2} \tag{3.29}$$

Thus, merging terms with the same denominator, we get in total

$$\frac{f}{g} = \frac{2}{q_1 \cdot q_3 \cdot q_4} + \frac{-1}{q_1 \cdot q_2 \cdot q_4} + \frac{-2}{q_3 \cdot q_4 \cdot q_5} + \frac{3x_4}{q_3 \cdot q_4 \cdot q_5^2} + \frac{-3x_4}{q_1 \cdot q_3 \cdot q_5^2} \tag{3.30}$$

where all terms satisfy conditions (1) to (3).

Example for syzygy reduction: If we had done the syzygy reduction step, for the first step in the short numerator decomposition above, i.e. the decomposition

$$\frac{f}{g} = \frac{x_2 + x_3 - x_4}{q_1 \cdot q_2 \cdot q_3 \cdot q_4} + \frac{-2x_1 + 3x_4}{q_1 \cdot q_3 \cdot q_4 \cdot q_5} \tag{3.31}$$

arising from the division expression

$$f = (-2x_1 + 3x_4) \cdot q_2 + (x_2 + x_3 - x_4) \cdot q_5, \tag{3.32}$$

we would have to calculate a Gröbner basis of the syzygy module of the ideal $\langle q_1, q_2, q_3, q_4, q_5 \rangle$. Using Singular we get the reduced Gröbner basis

$$\begin{aligned} \{\, &(-1, 0, 0, 1, 1)\,, (-x_3, x_5, 0, x_3, 0)\,, (0, x_2 - x_3 - x_4, x_3, 0, 0)\,, \\ &(-x_2 + x_4, x_5, -x_5, x_2 - x_4, 0)\,, (-x_3, x_1, 0, 0, 0)\,, \\ &(x_2 - x_3 - x_4, 0, x_1, 0, 0)\,, (-x_1 + x_5, 0, 0, x_1, 0)\,\} \end{aligned} \tag{3.33}$$

with respect to the graded reverse lexicographic ordering with priority to monomials (see [68], Definition 2.3.1). The original relation (3.32) corresponds to the module element $(0, -2x_1 + 3x_4, 0, 0, x_2 + x_3 - x_4)$ and dividing by the Gröbner basis yields the remainder $(-2x_3, 3x_4 + 2x_5, -x_5, 0, 0)$. So we would use the relation

$$f = (-2x_3) \cdot q_1 + (3x_4 + 2x_5) \cdot q_2 + (-x_5) \cdot q_3 \tag{3.34}$$

instead of relation (3.32), leading to the decomposition step

$$\frac{f}{g} = \frac{-2x_3}{q_2 \cdot q_3 \cdot q_4 \cdot q_5} + \frac{3x_4 + 2x_5}{q_1 \cdot q_3 \cdot q_4 \cdot q_5} + \frac{-x_5}{q_1 \cdot q_2 \cdot q_4 \cdot q_5}. \tag{3.35}$$

This simple example may not show, why syzygy reduction should be an advantage, since here the decomposition seems to get longer, but, as mentioned above, for more complicated input functions we observe a significant improvement of the performance of the algorithm.

4 Examples

In the following, we discuss the application of the partial fraction decomposition to IBP reduction coefficients of Feynman integrals, in examples of various complexity, and with and without use of a UT basis.

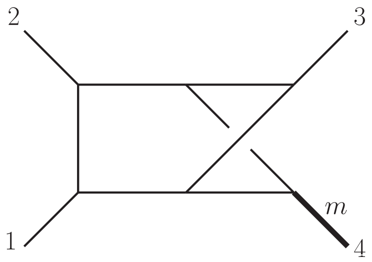

Figure 1. One-massive crossed box.

4.1 A baby example: nonplanar two-loop four-point with an external massive leg

In this subsection, we present a baby example, one-massive crossed box, showing how partial fraction decomposition simplifies the IBP reduction coefficients. The diagram is shown in figure 1.

The physical kinematic conditions are that p_1, p_2 and p_3 are massless, while $p_4^2 = m^2$, $2p_1 \cdot p_2 = s$ and $2p_2 \cdot p_3 = t$. The propagators are

$$\begin{aligned} &D_1 = k_1^2,\ D_2 = (k_1 - p_1)^2,\ D_3 = (k_1 - p_1 - p_2)^2,\ D_4 = k_2^2, \\ &D_5 = (k_2 + p_1 + p_2 + p_3)^2,\ D_6 = (k_2 + k_1 + p_3)^2,\ D_7 = (k_1 + k_2)^2, \\ &D_8 = (k_1 - p_1 - p_2 - p_3)^2,\ D_9 = (k_2 + p_1)^2 \end{aligned} \tag{4.1}$$

The parameters are thus $\epsilon = (4-d)/2$, s, t, and m^2.

We study the IBP reduction coefficients of integrals in the sector $(1,1,1,1,1,1,1,0,0)$ with the ISP degrees up to 5. This is a simple example, the IBP reduction can be easily done with LiteRed/FIRE6 [9, 15]. There are 29 master integrals and the evaluation of the master integrals was done by Gehrmann and Remiddi in ref. [71]. With LiteRed/FIRE6's master integral choice, the byte size of the IBP reduction coefficients is around 9.5MB.

We discuss the coefficients in more detail, listing the irreducible denominator factors (poles) below:

$$\begin{array}{llllll} \epsilon+1, & -2\epsilon-1, & -2\epsilon, & 1-2\epsilon, & 2-2\epsilon, & 3-2\epsilon, \\ 2-\epsilon, & -4\epsilon-1, & 1-4\epsilon, & 3-4\epsilon, & 1-3\epsilon, & 2-3\epsilon, \\ m^2, & m^2-s, & s, & \multicolumn{3}{l}{-10m^2\epsilon-6m^2+12s\epsilon+8s,} \\ m^2-t, & m^2-s-t, & t, & s+t, & \multicolumn{2}{l}{m^2s-m^2t-s^2-st,} \end{array} \tag{4.2}$$

It is not surprising that there is a pole $-10m^2\epsilon - 6m^2 + 12s\epsilon + 8s$, with the dependence in both ϵ and the kinematic parameters. There is also a nonlinear pole, $m^2s - m^2t - s^2 - st$ occurring in the list above.

We then convert the IBP reduction coefficients to a UT basis. It is easy to find the UT basis via leading singularity analysis or Wasser's dlog algorithm [57, 72]. The IBP reduction coefficients of the UT basis clearly have simpler poles:

$$\begin{array}{llllllll} \epsilon-2, & \epsilon-1, & 2\epsilon-3, & 2\epsilon-1, & 3\epsilon-2, & 3\epsilon-1, & 4\epsilon-3, & 4\epsilon-1, \\ m^2, & m^2-s, & s, & m^2-t, & m^2-s-t, & t, & s+t & \end{array} \tag{4.3}$$

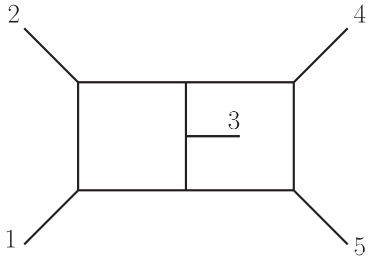

Figure 2. 2-loop 5-point nonplanar double pentagon.

We find the previously occurring factor $-10m^2\epsilon - 6m^2 + 12s\epsilon + 8s$ with mixed dependence in ϵ and kinematic variables is now absent. Furthermore, all the kinematic dependent poles are symbol letters, as seen by a comparison with the canonical differential equation. The previously occurring denominator factor $m^2s - m^2t - s^2 - st$, which is *not* a symbol letter, is also absent.

Note that in this example, the size of the IBP coefficients with respect to the UT basis, is around 9.0MB. By converting to the UT basis, the byte size of coefficients does not decrease much, but the denominator structure becomes much simpler.

We then apply our implementation of our partial fractioning algorithm to the IBP reduction coefficients, both with respect to the Laporta and the UT basis, to simplify the coefficients.

- After applying the algorithm, the size of IBP coefficients with respect to the Laporta basis (LiteRed/FIRE6) is shortened from 9.5MB to 3.0MB (2.7MB if indexed), simplified by about a factor of 3.4.
- Converting to the UT basis and then applying the algorithm, the resulting coefficients are only of size 1.9 MB (1.5 MB if indexed). With respect to the original Laporta basis a 6.5-times size reduction.

This example indicates that our method works for both the Laporta basis and UT basis, but the size reduction ratio is larger for UT basis. Since this is a baby example, our method runs fast in both cases.

4.2 A cutting-edge example: nonplanar two-loop five-point

In this section, we present a computationally cutting-edge example, the two-loop five-point nonplanar double pentagon. The diagram is shown in figure 2. All external and internal lines are massless. The kinematic conditions are $2p_1 \cdot p_2 = s_{12}$, $2p_2 \cdot p_3 = s_{23}$, $2p_3 \cdot p_4 = s_{34}$, $2p_4 \cdot p_5 = s_{45}$ and $2p_1 \cdot p_5 = s_{15}$. The propagators are

$$\begin{aligned} &D_1 = l_1^2, && D_2 = (l_1 - p_1)^2, && D_3 = (l_1 - p_{12})^2, && D_4 = l_2^2, \\ &D_5 = (l_2 - p_{123})^2, && D_6 = (l_2 - p_{1234})^2, && D_7 = (l_1 - l_2)^2, && \\ &D_8 = (l_1 - l_2 + p_3)^2, && D_9 = (l_1 - p_{1234})^2, && D_{10} = (l_2 - p_1)^2, && \\ &D_{11} = (l_2 - p_{12})^2. &&&&&& \end{aligned} \tag{4.4}$$

Where $p_{i\dots j} = \sum_{k=i}^{j} p_k$. A UT basis for this diagram and its symbol form was found in refs. [58, 73], and the analytic expressions for the master integrals were obtained in ref. [58].

Relying on the module intersection IBP reduction method and its implementation in the Singular-GPI-Space framework for massively parallel computations, the analytic IBP reduction coefficients were calculated for the integrals with ISP up to the degree 4 in the sector $(1,1,1,1,1,1,1,1,0,0,0)$ in ref. [33].[2] The size of the IBP reduction coefficients with respect to a Laporta basis is 2.4GB (with all parameters analytic).

When reducing target integrals to the Laporta basis, we found some "mixed" denominator factors in the coefficients, which are mixtures of the spacetime parameter ϵ and kinematic variables. They are listed in the following:

$$\begin{aligned}
&3s_{12}\epsilon - s_{23}\epsilon - s_{12} + s_{23},\\
&3s_{12}\epsilon - s_{23}\epsilon - 4s_{34}\epsilon - 3s_{45}\epsilon + s_{23} + s_{34},\\
&3s_{12}\epsilon + 4s_{23}\epsilon + s_{34}\epsilon - 3s_{45}\epsilon - s_{23} - s_{34},\\
&3s_{12}\epsilon + 8s_{23}\epsilon - 3s_{45}\epsilon - s_{12} - 4s_{23} + s_{45},\\
&4s_{12}\epsilon + s_{23}\epsilon - s_{45}\epsilon - 2s_{12} - s_{23} + s_{45}
\end{aligned} \tag{4.5}$$

As we have observed in ref. [33], if we reduce the target integrals to the UT basis, the size of coefficients is reduced to 712MB. More importantly, the IBP reduction coefficients with respect to UT basis have no "mixed" poles and all kinematic denominators are symbol letters. The irreducible factors in the IBP reduction coefficients with respect to the UT basis are given as,

$$\begin{aligned}
&\epsilon-1,\quad 2\epsilon-1,\quad 3\epsilon-1,\quad 4\epsilon-1,\quad 4\epsilon+1,\quad s_{12},\quad s_{15},\quad s_{15}-s_{23},\\
&s_{23},\quad s_{12}+s_{23},\quad s_{12}-s_{34},\quad s_{12}+s_{15}-s_{34},\quad s_{15}-s_{23}-s_{34},\quad s_{34},\\
&s_{23}+s_{34},\quad s_{12}-s_{45},\quad s_{23}-s_{45},\quad s_{12}+s_{23}-s_{45},\quad s_{12}-s_{15}+s_{23}-s_{45},\\
&s_{12}-s_{34}-s_{45},\quad s_{12}+s_{15}-s_{34}-s_{45},\quad s_{45},\quad s_{15}-s_{23}+s_{45},\quad s_{34}+s_{45},\\
&s_{15}^2s_{12}^2+s_{23}^2s_{12}^2-2s_{15}s_{23}s_{12}^2-2s_{23}^2s_{34}s_{12}+2s_{15}s_{23}s_{34}s_{12}-2s_{15}^2s_{45}s_{12}\\
&\;+2s_{15}s_{23}s_{45}s_{12}+2s_{15}s_{34}s_{45}s_{12}+2s_{23}s_{34}s_{45}s_{12}+s_{23}^2s_{34}^2+s_{15}^2s_{45}^2+s_{34}^2s_{45}^2\\
&\;-2s_{15}s_{34}s_{45}^2-2s_{23}s_{34}^2s_{45}+2s_{15}s_{23}s_{34}s_{45}
\end{aligned} \tag{4.6}$$

We see that except for the factors only in ϵ, all other factors are (powers of) even symbol letters. (The symbol letters of all two-loop five-point massless topology were obtained in ref. [74].) Note that the last factor above is the Gram determinant $G(1,2,3,4)$. Since

$$G(1,2,3,4) = \epsilon_5^2 \equiv -16(\epsilon_{\mu\nu\rho\sigma}k_1^\mu k_2^\nu k_3^\rho k_4^\sigma)^2 \tag{4.7}$$

and ϵ_5 is a symbol letter, the last factor $G(1,2,3,4)$ is a power of a symbol letter. In addition to the computation in ref. [33], we further checked the IBP reduction of the integrals with ISP up to the degree 5 for the same diagram, and this pole structure property still holds.

[2] We note that during the preparation of this manuscript, the degree-5 analytic IBP reduction of the same diagram, was calculated in a recent paper [16] by the reduction of a new type of IBP system in the block triangular form [40].

At this point, it is interesting to compare the pole structure in the UT basis with the same double pentagon diagram reduced in the basis choice of ref. [45]. In ref. [45] there are 7 nonlinear irreducible factors in the IBP reduction denominators, such as $-1+s_{34}+s_{45}+s_{34}s_{45}$, $-s_{23}-s_{45}-s_{23}s_{45}+s_{45}^2$, $s_{15}-s_{23}+s_{23}s_{34}-s_{15}s_{45}+s_{34}s_{45}$, ... ($s_{12}$ is set to be 1). Except the Gram determinant $G(1,2,3,4)$, the other 6 nonlinear factors in ref. [45] are not symbol letters.

Despite the fact that the size of the coefficients is already simplified by about 3 times, if we change the basis from Laporta to UT, the coefficients are still huge with a size of 712MB. We now apply our improved Leinartas' algorithm to shorten these coefficients with respect to the UT basis. The size of the coefficients is magically shortened to only 24MB (19MB in indexed form). Compared with the 2.4GB IBP reduction file we started out with, those IBP reduction coefficients are made simpler by over 100 times![3]

As a comparison, without using the UT basis, our algorithm can also reduce the IBP coefficients size from 2.4 G to 864 MB. However, the reduction ratio is not as dramatic and the running time is much longer.

In appendix B we present a visual impression about how powerful our algorithm is: a 5-page-long coefficient is shortened to only 9 lines.

4.3 An elliptic example: two-loop four-point with a top quark loop and a pair of external massive legs

Our algorithm also works well for cases without the existence of a UT basis. In this subsection, we present an elliptic example, the double box diagram with one massive internal loop and two massive external lines. The diagram is shown in figure 3.

The kinetic conditions are $p_1^2=p_2^2=0$, $p_3^2=m^2$, $2p_1\cdot p_2=s$, $2p_2\cdot p_3=t-m^2$ and $2p_2\cdot p_3=m^2-s-t$. The propagators are

$$\begin{array}{lll} D_1=(l_1+p_2)^2-m_t^2, & D_2=l_1^2-m_t^2, & D_3=(l_1+p_1+p_2)^2-m_t^2, \\ D_4=(l_1+l_2)^2-m_t^2, & D_5=l_2^2, & D_6=(l_2-p_1-p_2)^2, \quad D_7=(l_2+p_3)^2-m^2, \\ D_8=(l_1+p_2-p_3)^2-m^2, & D_9=(l_2-p_2+p_3)^2 & \end{array} \tag{4.8}$$

The parameters are $\epsilon, s, t, m^2, m_t^2$. It is clear that there are fully massive sunset subdiagrams in this topology and the UT basis does not exist.

We have reduced integrals in the sector $(1,1,1,1,1,1,1,0,0)$, with the ISP degree up to 5 to the Laporta basis, using FIRE6. The size of the resulting coefficients is in total 175 MB.

In applying our algorithm to shorten these coefficients, it is important to pull the nonlinear factors out and do the partial fractions over the linear factors. After applying our algorithm, the size of simplified coefficients is reduced to only 24 MB. This is also a significant simplification, by about 7 times in byte-size.

This example indicates that although one should prefer a UT basis in doing partial fraction, for diagrams without the existence of UT basis, this algorithm is still powerful.

[3] The resulting indexed partial fraction result can be downloaded from https://www.dropbox.com/s/wku5o20g0vaggtl/xb_deg4_UT_pfd.zip.

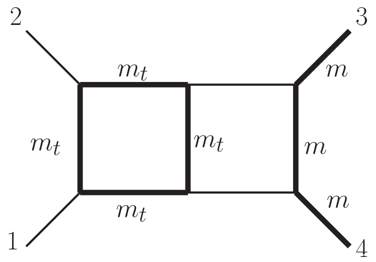

Figure 3. Elliptic double box.

example	sequential		parallel (32 cores)		memory for just reading input file
	runtime	memory	runtime	memory	
⋄ xbox1m (Laporta)	929 s	67 MB	74 s	171 MB	38 MB
xbox1m (UT)	1073 s	53 MB	48 s	151 MB	30 MB
⋄ dpentagon (Laporta)	–	–	137 h	44.3 GB	9.63 GB
dpentagon (UT)	1672 min	5.39 GB	99 min	10.24 GB	2.45 GB
⋄ dbox elliptic (Laporta)	304 min	1.24 GB	41 min	2.18 GB	495 MB

Table 1. Runtime and memory used in algorithm 5.

4.4 Performance of the algorithm

In this section, we summarize the computing resources used for our examples, and the reduction ratio in different formats.

In all examples a Singular implementation of algorithm 5 with the improvements described in Remarks 3.11 and 3.13 was used. Table 1 shows the resources used for applying the algorithm to all matrix entries one after the other or in parallel using 32 cores. Due to the simple form of parallelism, the computation will scale similarly up to the number of entries.

When comparing the time taken for each decomposition step, we found that the short numerator decomposition (algorithm 4) needs 60–95 % of the total runtime.

Especially for large numerators and small (by degree) denominator factors, partial fraction decomposition can drastically reduce the size of IBP-matrices, as can be seen in table 2. Since most of the irreducible factors in the denominators are linear, it makes sense to leave any nonlinear factors untouched in the algorithm. In the tables in this subsection, the symbol "⋄" means that we leave the nonlinear factors untouched in our partial fraction algorithm. The phrase "(Laporta)" or "(UT)" means that we are dealing with coefficients in a Laporta integral basis or a UT basis, respectively.

The use of a UT basis typically leads to a shorter runtime (table 1) and also reduces the size of the output (table 2). Finally, instead of writing out the denominator, we can just store in the data structure the indices i and exponents e_i of the irreducible factors q_i appearing in each denominator together with all factors q_i, which also reduces the size (last column in table 2).

example	input	output	output (indexed)
⋄ xbox1m (Laporta)	9.51 MB	2.91 MB (30.7%)	2.75 MB (29.0%)
xbox1m (UT)	9.04 MB	1.80 MB (19.9%)	1.53 MB (16.9%)
⋄ double pentagon (Laporta)	2.42 GB	864 MB (35.7%)	851 MB (35.2%)
double pentagon (UT)	712 MB	28.0 MB (3.93%)	19.8 MB (2.78%)
⋄ dbox elliptic (Laporta)	175 MB	24.1 MB (13.8%)	23.3 MB (13.3%)

Table 2. Size comparison of input/output of algorithm 5.

example	input (zipped)	output (zipped)	output (indexed, zipped)
⋄ xbox1m (Laporta)	2.90 MB	829 KB (28.6%)	815 KB (28.1%)
xbox1m (UT)	2.43 MB	369 KB (15.2%)	351 KB (14.5%)
⋄ double pentagon (Laporta)	648 MB	216 MB (33.3%)	215 MB (33.2%)
double pentagon (UT)	212 MB	4.42 MB (2.09%)	3.99 MB (1.89%)
⋄ dbox elliptic (Laporta)	50.5 MB	7.60 MB (15.0%)	7.51 MB (14.9%)

Table 3. Size comparison of input/output of algorithm 5 after zipping.

We also find an interesting phenomenon that zipping both the input and output files in some examples leads to a further increase of the relative size reduction (see table 3).

5 Summary and discussion

In this manuscript, we develop an improved Leinartas' algorithm of multivariate partial fraction and present an modern implement of this algorithm, to simplify the complicated analytic IBP reduction coefficients in multi-loop computations. We show that for cases with the existence or without the existence of the UT basis, our algorithm works well to reduce the IBP reduction coefficients size.

We observe that in the cases we studied, the IBP reduction coefficients in the UT basis have simple structures: (1) the spacetime dimension parameter D factorizes out in the denominators (2) the rest irreducible factors in the denominators are a subset of the symbol letters. Thus usually the UT basis provides a simpler denominator factor list, and our algorithm works particularly well with shorter running time and higher reduction ratio. In complicated examples, our algorithm achieves dramatic size reduction in the coefficients of IBPs.

We expect that our algorithm will have broad applications in the multi-loop IBP computations, to get easier-to-use analytic reduction results and make the numeric evaluation much faster.

We present a Singular library for our algorithm of multivariate partial fraction. It can be used for simplifying IBP coefficients in general purposes. Furthermore, we expect that the partial-fraction library can be used to simplify multiloop integrand and the transcendental function coefficients in scattering amplitudes, as the partial-fraction examples shown in [51]. We expect that this library can be combined with current finite field and rational reconstruction packages [9, 27, 29–31] for multiloop scattering amplitude computations. Our library would also find applications in analytic computations outside scattering amplitudes, in broader research areas in theoretical physics.

Besides this partial fraction library, in the future we will also develop an arithmetic library to perform arithmetic computations of ration functions in partial fraction form and keep the output in partial fraction form.

It would also be interesting to study the IBP reduction coefficients in a UT basis in details. After partial fraction, it seems that each team in a coefficient looks much simpler. It is then of theoretical interests to relate these terms to the leading singularities of Feynman integrals.

Acknowledgments

We acknowledge Johannes Henn for enlightening discussions and his valuable comments on our original manuscript. We are also grateful to David Kosower, Kasper Larsen, Roman Lee, Hui Luo, Yanqing Ma, Andreas von Manteuffel, Erik Panzer, Robert Schabinger, Alexander Smirnov, Vladimir Smirnov, Johann Usovitsch and Pascal Wasser for very helpful discussions on related topics.

The work of JB was supported by Project II.5 of SFB-TRR 195 Symbolic Tools in Mathematics and their Application” of the German Research Foundation (DFG). The work of YZ was support from the NSF of China through Grant No. 11947301.

A Manual of the partial fractioning Singular library

In this section, we give a short outline of how to use the features of the Singular library `pfd.lib`. Together with a complete documentation, it can be downloaded from

https://raw.githubusercontent.com/Singular/Singular/spielwiese/Singular/LIB/pfd.lib

and should be placed within the user's Singular search path. The latest release of Singular can be downloaded from the Singular website https://www.singular.uni-kl.de.[4] The website also provides an online documentation of Singular and all libraries distributed with the release.

After starting up Singular, the library can be loaded by typing `LIB "pfd.lib";` at the Singular promt. The main algorithm for partial fraction decomposition can be accessed via the procedure `pfd`. This procedure takes as input two polynomials (numerator and denominator) and returns a partial fraction decomposition encoded as a list. The first entry is a list containing the denominator factors, the second entry is a list of summands, each of which is encoded as a list of numerator, indices of denominator factors and exponents of denominator factors. The decomposition can be displayed using the procedure `displaypfd`, and checked with the procedure `checkpfd`, which verifies whether a rational function (first argument) is mathematically equal to a decomposition returned by `pfd`

[4] A separate download will not be necessary in the future, since the library will be part of the next release of Singular (version 4.1.4).

(second argument) and returns a boolean (see Example A.1). An example of how to use a procedure can be displayed by typing `example <name-of-procedure>;` at the Singular prompt.

Example A.1 (`pfd`)**.** In the following example, we calculate a partial fraction decomposition of the rational function $(x^2+3xy-y^2)/((x+1)\cdot(2x+y)) \in \mathbb{Q}(x,y)$ with respect to the graded reverse lexicographic ordering:

```
> LIB "pfd.lib";
> ring r = 0,(x,y),dp;
> poly f = x^2+3xy-y^2;
> poly g = (x+1)*(2x+y);
> list d = pfd(f,g);
> displaypfd(d);
  (1/2)
+ (9/2y+9) / (q2)
+ (-y-5) / (q1)
+ (-9) / (q1*q2)
where
q1 = x+1
q2 = 2x+y

> checkpfd(list(f,g),d);
1
```

In the first line we create the underlying polynomial ring by specifying the characteristic of the coefficient field (`0` for $\mathbb{Q}$), the names of the variables and a monomial ordering (`dp` for the graded reverse lexicographic ordering). Note that the result produced by the algorithm depends on the ordering.

The second argument (the denominator polynomial) can alternatively be given in factorized form (as a list of a Singular `ideal` generated by irreducible non-constant polynomials and an `intvec` containing the exponents), in case the denominator factors are known to the user. As an example

```
> pfd(x+2*y, (x+y)^2*(x-y)^3);
```

is equivalent to

```
> pfd(x+2*y, list(ideal(x+y,x-y), intvec(2,3)));
```

Using the procedure `pfdMat`, we can calculate the decompositions of a matrix of rational functions. The computation is done in parallel, relying on the library `parallel.lib`.[5] By default, `pfdMat` also calls `checkpfd` for each decomposition and ignores nonlinear denominator factors (as described in section 4).

The input of `pfdMat` is the name of a `.txt`-file (as a Singular `string`), which contains the matrix as a list of lists (row by row) enclosed in the symbols "{" and "}", and separated

[5] A version relying on parallelism implemented via the Singular/GPI-Space framework [75] is under development. This version will allow the use of our algorithm on HPC clusters.

by commas (see Example A.2). Each rational function has to be an expression of the form “`a`”, “`(a)/(b)`”, “`(b)^(-n)`” or “`(a)*(b)^(-n)`”, where “`n`” stands for a positive integer and “`a`”, “`b`” stand for arbitrary polynomials (using the operators “`+`”, “`-`”, “`*`”, “`^`” and brackets “`(`”,“`)`”). A minus sign “`-`” followed by such an expression is also allowed. Note that the library also has options to use the Singular binary serialization data format `.ssi` for highly efficient input and output from within Singular.

There are four optional arguments which determine whether `checkpfd` should be applied (`-1`: exact test, `0`: do not apply `checkpfd`, positive integer: do this amount of probabilistic tests, default value is `-1`), whether nonlinear factors should be extracted (`1` or `0`, default value `1`), whether additional output files should be created (integer from `1` to `4`, default value `1`) and whether the algorithm should be run in parallel over all matrix entries (`1` or `0`, default value `1`). The options should be specified in this order. The third optional argument (integer from `1` to `4`) controls the output files created:

`1`: The output, that is, the matrix containing the decompositions, is stored in a `.txt`-file in indexed form (as described in section 4). The denominator factors are saved in a separate file and a logfile is created, which protocols runtimes and memory usage.

`2`: Additionally, the decompositions are saved in non-indexed form.

`3`: Additional `.ssi`-files containing the input and output matrix as well as some intermediate results are created.

`4`: Additionally to mode `3`, for every rational function, the result of `pfd` is immediately saved in a seperate `.ssi`-file. (This creates a file for every matrix entry.)

For more details refer to the documentation of the library.

Before calling `pfdMat`, a polynomial ring must be defined (as in Example A.1) such that the variable names match the names used in the input file. Furthermore, with the command `setcores(n);` the number of processor cores used for the parallelization can be set to an integer `n`. By default, all cores are used.

Example A.2 (`pfdMat`)**.** Suppose the file `test.txt` in the search path contains the string “`{{(x+y)/(x^2-x*y),-(x^2*y+1)/y,x^2},{(x+y+1)/(y^2),0,(x^2*y-y^3)^(-1)}}`” representing a 2×3 matrix of rational functions. We then can calculate the decompositions of the 6 matrix entries (in parallel on 4 cores) as follows:

```
> LIB "pfd.lib";
> setcores(4);
> ring r = 0,(x,y),dp;
> pfdMat("test.txt");
```

The procedure `pfdMat` creates two output files: `test_pfd_indexed.txt` containing the matrix of partial fraction decompositions as the string

```
“{{(2)/(q2) + (-1)/(q1), (-x^2) + (-1)/(q3), (x^2)},
 {(1)/(q3) + (x+1)/(q3^2), (0), (1)/(q3*q4^2) + (2)/(q2*q4^2)}}”
```

and `test_denominator_factors.txt` containing the factors `q1`, `q2`, `q3`, `q4` in the form

“`q1 = x;   q2 = x-y;   q3 = y;   q4 = x+y;`”.

As explained above, `pfdMat` has four optional arguments with default values `-1,1,1,1`. If the third argument is set to 2 as in “`pfdMat("test.txt",-1,1,2,1);`”, an additional file `test_pfd.txt` is created, which contains the result in non indexed form as the string

```
“{{(2)/((x-y)) + (-1)/((x)), (-x^2) + (-1)/((y)), (x^2)},
{(1)/((y)) + (x+1)/((y)^2), (0), (1)/((y)*(x+y)^2) + (2)/((x-y)*(x+y)^2)}}”.
```

Remark A.3. In an IBP-matrix, typically the total number of distinct irreducible factors occurring in the denominators of all the rational functions is fairly small (e.g. for the double pentagon example in UT-basis, there are only 25 different irreducible denominator factors occurring in the $26 \times 108 = 2808$ matrix entries). The procedure `pfdMat` uses this fact to speed up the factorization of the denominators by dividing by factors obtained from factorizing “simpler” polynomials in order to factorize the larger polynomials.

B An explicit example of the size reduction

In this appendix, we explicitly show an IBP reduction coefficient, before and after our partial fraction computations to see the size reduction.

This example is from the 2-loop 5-point nonplanar double pentagon IBP reduction (figure. 2). We choose a UT basis I_i, $i = 1, \ldots, 108$. The integral to be reduced is $I[1,1,1,1,1,1,1,1,-2,-1,0]$,

$$I[1,1,1,1,1,1,1,1,-2,-1,0] = \sum_{i=1}^{108} c_i I_i \,. \tag{B.1}$$

Because of the UT basis, the coefficients shown in (B.1) are simpler than the coefficients in FIRE basis. However, the coefficients are still very complicated. We pick up the particular coefficient c_{107} as an example. Before the partial fraction, in the usually combined form, we have a huge expression of c_{107}. It is can listed in about five pages as:

$$\begin{aligned}
c_{107} = \Big(&- 64\epsilon s_{15}{}^3 s_{45}{}^7 - 16 s_{15}{}^3 s_{45}{}^7 + 64\epsilon s_{34}{}^3 s_{45}{}^7 + 16 s_{34}{}^3 s_{45}{}^7 - 4\epsilon s_{12} s_{15}{}^2 s_{45}{}^7 \\
&- s_{12} s_{15}{}^2 s_{45}{}^7 - 4\epsilon s_{12} s_{34}{}^2 s_{45}{}^7 - s_{12} s_{34}{}^2 s_{45}{}^7 - 192\epsilon s_{15} s_{34}{}^2 s_{45}{}^7 - 48 s_{15} s_{34}{}^2 s_{45}{}^7 \\
&- 48\epsilon s_{23} s_{34}{}^2 s_{45}{}^7 - 12 s_{23} s_{34}{}^2 s_{45}{}^7 - 48\epsilon s_{15}{}^2 s_{23} s_{45}{}^7 - 12 s_{15}{}^2 s_{23} s_{45}{}^7 + 12\epsilon s_{12} s_{15} s_{23} s_{45}{}^7 \\
&+ 3 s_{12} s_{15} s_{23} s_{45}{}^7 + 192\epsilon s_{15}{}^2 s_{34} s_{45}{}^7 + 48 s_{15}{}^2 s_{34} s_{45}{}^7 + 8\epsilon s_{12} s_{15} s_{34} s_{45}{}^7 + 2 s_{12} s_{15} s_{34} s_{45}{}^7 \\
&- 12\epsilon s_{12} s_{23} s_{34} s_{45}{}^7 - 3 s_{12} s_{23} s_{34} s_{45}{}^7 + 96\epsilon s_{15} s_{23} s_{34} s_{45}{}^7 + 24 s_{15} s_{23} s_{34} s_{45}{}^7 - 64\epsilon s_{15}{}^4 s_{45}{}^6 \\
&- 16 s_{15}{}^4 s_{45}{}^6 + 128\epsilon s_{34}{}^4 s_{45}{}^6 + 32 s_{34}{}^4 s_{45}{}^6 + 248\epsilon s_{12} s_{15}{}^3 s_{45}{}^6 + 62 s_{12} s_{15}{}^3 s_{45}{}^6 \\
&- 196\epsilon s_{12} s_{34}{}^3 s_{45}{}^6 - 49 s_{12} s_{34}{}^3 s_{45}{}^6 - 320\epsilon s_{15} s_{34}{}^3 s_{45}{}^6 - 80 s_{15} s_{34}{}^3 s_{45}{}^6 - 352\epsilon s_{23} s_{34}{}^3 s_{45}{}^6 \\
&- 76 s_{23} s_{34}{}^3 s_{45}{}^6 + 12\epsilon s_{12}{}^2 s_{15}{}^2 s_{45}{}^6 + 3 s_{12}{}^2 s_{15}{}^2 s_{45}{}^6 - 12\epsilon s_{12}{}^2 s_{23}{}^2 s_{45}{}^6 - 3 s_{12}{}^2 s_{23}{}^2 s_{45}{}^6 \\
&+ 96\epsilon s_{15}{}^2 s_{23}{}^2 s_{45}{}^6 + 24 s_{15}{}^2 s_{23}{}^2 s_{45}{}^6 - 96\epsilon s_{12} s_{15} s_{23}{}^2 s_{45}{}^6 - 24 s_{12} s_{15} s_{23}{}^2 s_{45}{}^6 \\
&+ 8\epsilon s_{12}{}^2 s_{34}{}^2 s_{45}{}^6 + 2 s_{12}{}^2 s_{34}{}^2 s_{45}{}^6 + 192\epsilon s_{15}{}^2 s_{34}{}^2 s_{45}{}^6 + 48 s_{15}{}^2 s_{34}{}^2 s_{45}{}^6 \\
&+ 144\epsilon s_{23}{}^2 s_{34}{}^2 s_{45}{}^6 + 36 s_{23}{}^2 s_{34}{}^2 s_{45}{}^6 + 640\epsilon s_{12} s_{15} s_{34}{}^2 s_{45}{}^6 + 160 s_{12} s_{15} s_{34}{}^2 s_{45}{}^6 \\
&+ 264\epsilon s_{12} s_{23} s_{34}{}^2 s_{45}{}^6 + 54 s_{12} s_{23} s_{34}{}^2 s_{45}{}^6 + 784\epsilon s_{15} s_{23} s_{34}{}^2 s_{45}{}^6 + 172 s_{15} s_{23} s_{34}{}^2 s_{45}{}^6 \\
&+ 80\epsilon s_{15}{}^3 s_{23} s_{45}{}^6 + 20 s_{15}{}^3 s_{23} s_{45}{}^6 + 176\epsilon s_{12} s_{15}{}^2 s_{23} s_{45}{}^6 + 44 s_{12} s_{15}{}^2 s_{23} s_{45}{}^6 \\
&- 24\epsilon s_{12}{}^2 s_{15} s_{23} s_{45}{}^6 - 6 s_{12}{}^2 s_{15} s_{23} s_{45}{}^6 + 64\epsilon s_{15}{}^3 s_{34} s_{45}{}^6 + 16 s_{15}{}^3 s_{34} s_{45}{}^6 \\
&- 692\epsilon s_{12} s_{15}{}^2 s_{34} s_{45}{}^6 - 173 s_{12} s_{15}{}^2 s_{34} s_{45}{}^6 - 12\epsilon s_{12} s_{23}{}^2 s_{34} s_{45}{}^6 - 3 s_{12} s_{23}{}^2 s_{34} s_{45}{}^6 \\
&- 240\epsilon s_{15} s_{23}{}^2 s_{34} s_{45}{}^6 - 60 s_{15} s_{23}{}^2 s_{34} s_{45}{}^6 - 20\epsilon s_{12}{}^2 s_{15} s_{34} s_{45}{}^6 - 5 s_{12}{}^2 s_{15} s_{34} s_{45}{}^6 \\
&+ 28\epsilon s_{12}{}^2 s_{23} s_{34} s_{45}{}^6 + 7 s_{12}{}^2 s_{23} s_{34} s_{45}{}^6 - 512\epsilon s_{15}{}^2 s_{23} s_{34} s_{45}{}^6 - 116 s_{15}{}^2 s_{23} s_{34} s_{45}{}^6
\end{aligned}$$

$$
\begin{aligned}
&-440\epsilon s_{12}s_{15}s_{23}s_{34}s_{45}^{6}-98s_{12}s_{15}s_{23}s_{34}s_{45}^{6}+64\epsilon s_{34}^{5}s_{45}^{5}+16s_{34}^{5}s_{45}^{5}\\
&+252\epsilon s_{12}s_{15}^{4}s_{45}^{5}+63s_{12}s_{15}^{4}s_{45}^{5}-376\epsilon s_{12}s_{34}^{4}s_{45}^{5}-94s_{12}s_{34}^{4}s_{45}^{5}-64\epsilon s_{15}s_{34}^{4}s_{45}^{5}\\
&-16s_{15}s_{34}^{4}s_{45}^{5}-560\epsilon s_{23}s_{34}^{4}s_{45}^{5}-116s_{23}s_{34}^{4}s_{45}^{5}-360\epsilon s_{12}^{2}s_{15}^{3}s_{45}^{5}\\
&-90s_{12}^{2}s_{15}^{3}s_{45}^{5}+24\epsilon s_{12}^{2}s_{23}^{3}s_{45}^{5}+6s_{12}^{2}s_{23}^{3}s_{45}^{5}-48\epsilon s_{15}^{2}s_{23}^{3}s_{45}^{5}\\
&-12s_{15}^{2}s_{23}^{3}s_{45}^{5}+156\epsilon s_{12}s_{15}s_{23}^{3}s_{45}^{5}+39s_{12}s_{15}s_{23}^{3}s_{45}^{5}+256\epsilon s_{12}^{2}s_{34}^{3}s_{45}^{5}\\
&+64s_{12}^{2}s_{34}^{3}s_{45}^{5}-192\epsilon s_{15}^{2}s_{34}^{3}s_{45}^{5}-48s_{15}^{2}s_{34}^{3}s_{45}^{5}+672\epsilon s_{23}^{2}s_{34}^{3}s_{45}^{5}\\
&+144s_{23}^{2}s_{34}^{3}s_{45}^{5}+1056\epsilon s_{12}s_{15}s_{34}^{3}s_{45}^{5}+264s_{12}s_{15}s_{34}^{3}s_{45}^{5}+1164\epsilon s_{12}s_{23}s_{34}^{3}s_{45}^{5}\\
&+231s_{12}s_{23}s_{34}^{3}s_{45}^{5}+1088\epsilon s_{15}s_{23}s_{34}^{3}s_{45}^{5}+236s_{15}s_{23}s_{34}^{3}s_{45}^{5}-12\epsilon s_{12}^{3}s_{15}^{2}s_{45}^{5}\\
&-3s_{12}^{3}s_{15}^{2}s_{45}^{5}+32\epsilon s_{12}^{3}s_{23}^{2}s_{45}^{5}+8s_{12}^{3}s_{23}^{2}s_{45}^{5}-32\epsilon s_{15}^{3}s_{23}^{2}s_{45}^{5}-8s_{15}^{3}s_{23}^{2}s_{45}^{5}\\
&-368\epsilon s_{12}s_{15}^{2}s_{23}^{2}s_{45}^{5}-92s_{12}s_{15}^{2}s_{23}^{2}s_{45}^{5}+236\epsilon s_{12}^{2}s_{15}s_{23}^{2}s_{45}^{5}+59s_{12}^{2}s_{15}s_{23}^{2}s_{45}^{5}\\
&-4\epsilon s_{12}^{3}s_{34}^{2}s_{45}^{5}-s_{12}^{3}s_{34}^{2}s_{45}^{5}+320\epsilon s_{15}^{3}s_{34}^{2}s_{45}^{5}+80s_{15}^{3}s_{34}^{2}s_{45}^{5}\\
&-144\epsilon s_{23}^{3}s_{34}^{2}s_{45}^{5}-36s_{23}^{3}s_{34}^{2}s_{45}^{5}-732\epsilon s_{12}s_{15}^{2}s_{34}^{2}s_{45}^{5}-183s_{12}s_{15}^{2}s_{34}^{2}s_{45}^{5}\\
&-832\epsilon s_{12}s_{23}^{2}s_{34}^{2}s_{45}^{5}-184s_{12}s_{23}^{2}s_{34}^{2}s_{45}^{5}-1152\epsilon s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{5}\\
&-240s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{5}-884\epsilon s_{12}^{2}s_{15}s_{34}^{2}s_{45}^{5}-221s_{12}^{2}s_{15}s_{34}^{2}s_{45}^{5}\\
&-464\epsilon s_{12}^{2}s_{23}s_{34}^{2}s_{45}^{5}-80s_{12}^{2}s_{23}s_{34}^{2}s_{45}^{5}-416\epsilon s_{15}^{2}s_{23}s_{34}^{2}s_{45}^{5}-104s_{15}^{2}s_{23}s_{34}^{2}s_{45}^{5}\\
&-2208\epsilon s_{12}s_{15}s_{23}s_{34}^{2}s_{45}^{5}-468s_{12}s_{15}s_{23}s_{34}^{2}s_{45}^{5}+80\epsilon s_{15}^{4}s_{23}s_{45}^{5}+20s_{15}^{4}s_{23}s_{45}^{5}\\
&-264\epsilon s_{12}s_{15}^{3}s_{23}s_{45}^{5}-66s_{12}s_{15}^{3}s_{23}s_{45}^{5}-220\epsilon s_{12}^{2}s_{15}^{2}s_{23}s_{45}^{5}-55s_{12}^{2}s_{15}^{2}s_{23}s_{45}^{5}\\
&+4\epsilon s_{12}^{3}s_{15}s_{23}s_{45}^{5}+s_{12}^{3}s_{15}s_{23}s_{45}^{5}-128\epsilon s_{15}^{4}s_{34}s_{45}^{5}-32s_{15}^{4}s_{34}s_{45}^{5}\\
&-200\epsilon s_{12}s_{15}^{3}s_{34}s_{45}^{5}-50s_{12}s_{15}^{3}s_{34}s_{45}^{5}+60\epsilon s_{12}s_{23}^{3}s_{34}s_{45}^{5}+15s_{12}s_{23}^{3}s_{34}s_{45}^{5}\\
&+192\epsilon s_{15}s_{23}^{3}s_{34}s_{45}^{5}+48s_{15}s_{23}^{3}s_{34}s_{45}^{5}+988\epsilon s_{12}^{2}s_{15}^{2}s_{34}s_{45}^{5}+247s_{12}^{2}s_{15}^{2}s_{34}s_{45}^{5}\\
&+128\epsilon s_{12}^{2}s_{23}^{2}s_{34}s_{45}^{5}+32s_{12}^{2}s_{23}^{2}s_{34}s_{45}^{5}+512\epsilon s_{15}^{2}s_{23}^{2}s_{34}s_{45}^{5}+104s_{15}^{2}s_{23}^{2}s_{34}s_{45}^{5}\\
&+796\epsilon s_{12}s_{15}s_{23}^{2}s_{34}s_{45}^{5}+175s_{12}s_{15}s_{23}^{2}s_{34}s_{45}^{5}+16\epsilon s_{12}^{3}s_{15}s_{34}s_{45}^{5}+4s_{12}^{3}s_{15}s_{34}s_{45}^{5}\\
&-20\epsilon s_{12}^{3}s_{23}s_{34}s_{45}^{5}-5s_{12}^{3}s_{23}s_{34}s_{45}^{5}-192\epsilon s_{15}^{3}s_{23}s_{34}s_{45}^{5}-36s_{15}^{3}s_{23}s_{34}s_{45}^{5}\\
&+1308\epsilon s_{12}s_{15}^{2}s_{23}s_{34}s_{45}^{5}+303s_{12}s_{15}^{2}s_{23}s_{34}s_{45}^{5}+632\epsilon s_{12}^{2}s_{15}s_{23}s_{34}s_{45}^{5}\\
&+122s_{12}^{2}s_{15}s_{23}s_{34}s_{45}^{5}-184\epsilon s_{12}s_{34}^{5}s_{45}^{4}-46s_{12}s_{34}^{5}s_{45}^{4}+64\epsilon s_{15}s_{34}^{5}s_{45}^{4}\\
&+16s_{15}s_{34}^{5}s_{45}^{4}-256\epsilon s_{23}s_{34}^{5}s_{45}^{4}-52s_{23}s_{34}^{5}s_{45}^{4}-372\epsilon s_{12}^{2}s_{15}^{4}s_{45}^{4}-93s_{12}^{2}s_{15}^{4}s_{45}^{4}\\
&-12\epsilon s_{12}^{2}s_{23}^{4}s_{45}^{4}-3s_{12}^{2}s_{23}^{4}s_{45}^{4}-72\epsilon s_{12}s_{15}s_{23}^{4}s_{45}^{4}-18s_{12}s_{15}s_{23}^{4}s_{45}^{4}\\
&+368\epsilon s_{12}^{2}s_{34}^{4}s_{45}^{4}+92s_{12}^{2}s_{34}^{4}s_{45}^{4}\quad-192\epsilon s_{15}^{2}s_{34}^{4}s_{45}^{4}-48s_{15}^{2}s_{34}^{4}s_{45}^{4}\\
&+912\epsilon s_{23}^{2}s_{34}^{4}s_{45}^{4}+180s_{23}^{2}s_{34}^{4}s_{45}^{4}+240\epsilon s_{12}s_{15}s_{34}^{4}s_{45}^{4}+60s_{12}s_{15}s_{34}^{4}s_{45}^{4}\\
&+1520\epsilon s_{12}s_{23}s_{34}^{4}s_{45}^{4}+308s_{12}s_{23}s_{34}^{4}s_{45}^{4}+208\epsilon s_{15}s_{23}s_{34}^{4}s_{45}^{4}+52s_{15}s_{23}s_{34}^{4}s_{45}^{4}\\
&+232\epsilon s_{12}^{3}s_{15}^{3}s_{45}^{4}+58s_{12}^{3}s_{15}^{3}s_{45}^{4}-52\epsilon s_{12}^{3}s_{23}^{3}s_{45}^{4}-13s_{12}^{3}s_{23}^{3}s_{45}^{4}\\
&+148\epsilon s_{12}s_{15}^{2}s_{23}^{3}s_{45}^{4}+37s_{12}s_{15}^{2}s_{23}^{3}s_{45}^{4}-384\epsilon s_{12}^{2}s_{15}s_{23}^{3}s_{45}^{4}-96s_{12}^{2}s_{15}s_{23}^{3}s_{45}^{4}\\
&-124\epsilon s_{12}^{3}s_{34}^{3}s_{45}^{4}-31s_{12}^{3}s_{34}^{3}s_{45}^{4}+192\epsilon s_{15}^{3}s_{34}^{3}s_{45}^{4}+48s_{15}^{3}s_{34}^{3}s_{45}^{4}\\
&-544\epsilon s_{23}^{3}s_{34}^{3}s_{45}^{4}-124s_{23}^{3}s_{34}^{3}s_{45}^{4}+572\epsilon s_{12}s_{15}^{2}s_{34}^{3}s_{45}^{4}+143s_{12}s_{15}^{2}s_{34}^{3}s_{45}^{4}\\
&-2284\epsilon s_{12}s_{23}^{2}s_{34}^{3}s_{45}^{4}-463s_{12}s_{23}^{2}s_{34}^{3}s_{45}^{4}-1392\epsilon s_{15}s_{23}^{2}s_{34}^{3}s_{45}^{4}\\
&-264s_{15}s_{23}^{2}s_{34}^{3}s_{45}^{4}-1160\epsilon s_{12}^{2}s_{15}s_{34}^{3}s_{45}^{4}-290s_{12}^{2}s_{15}s_{34}^{3}s_{45}^{4}\\
&-1668\epsilon s_{12}^{2}s_{23}s_{34}^{3}s_{45}^{4}-321s_{12}^{2}s_{23}s_{34}^{3}s_{45}^{4}+512\epsilon s_{15}^{2}s_{23}s_{34}^{3}s_{45}^{4}+92s_{15}^{2}s_{23}s_{34}^{3}s_{45}^{4}\\
&-2708\epsilon s_{12}s_{15}s_{23}s_{34}^{3}s_{45}^{4}-605s_{12}s_{15}s_{23}s_{34}^{3}s_{45}^{4}+4\epsilon s_{12}^{4}s_{15}^{2}s_{45}^{4}+s_{12}^{4}s_{15}^{2}s_{45}^{4}\\
&-28\epsilon s_{12}^{4}s_{23}^{2}s_{45}^{4}-7s_{12}^{4}s_{23}^{2}s_{45}^{4}+176\epsilon s_{12}s_{15}^{3}s_{23}^{2}s_{45}^{4}+44s_{12}s_{15}^{3}s_{23}^{2}s_{45}^{4}\\
&+588\epsilon s_{12}^{2}s_{15}^{2}s_{23}^{2}s_{45}^{4}+147s_{12}^{2}s_{15}^{2}s_{23}^{2}s_{45}^{4}-156\epsilon s_{12}^{3}s_{15}s_{23}^{2}s_{45}^{4}\\
&-39s_{12}^{3}s_{15}s_{23}^{2}s_{45}^{4}-64\epsilon s_{15}^{4}s_{34}^{2}s_{45}^{4}-16s_{15}^{4}s_{34}^{2}s_{45}^{4}+48\epsilon s_{23}^{4}s_{34}^{2}s_{45}^{4}\\
&+12s_{23}^{4}s_{34}^{2}s_{45}^{4}-1128\epsilon s_{12}s_{15}^{3}s_{34}^{2}s_{45}^{4}-282s_{12}s_{15}^{3}s_{34}^{2}s_{45}^{4}+920\epsilon s_{12}s_{23}^{3}s_{34}^{2}s_{45}^{4}\\
&+218s_{12}s_{23}^{3}s_{34}^{2}s_{45}^{4}+720\epsilon s_{15}s_{23}^{3}s_{34}^{2}s_{45}^{4}+156s_{15}s_{23}^{3}s_{34}^{2}s_{45}^{4}\\
&+832\epsilon s_{12}^{2}s_{15}^{2}s_{34}^{2}s_{45}^{4}+208s_{12}^{2}s_{15}^{2}s_{34}^{2}s_{45}^{4}+1508\epsilon s_{12}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{4}\\
&+317s_{12}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{4}+416\epsilon s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{4}+80s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{4}
\end{aligned}
$$

$$+ 2628\epsilon s_{12}s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{4} + 501s_{12}s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{4} + 560\epsilon s_{12}^{3}s_{15}s_{34}^{2}s_{45}^{4}$$

$$+ 140s_{12}^{3}s_{15}s_{34}^{2}s_{45}^{4} + 460\epsilon s_{12}^{3}s_{23}s_{34}^{2}s_{45}^{4} + 79s_{12}^{3}s_{23}s_{34}^{2}s_{45}^{4} - 624\epsilon s_{15}^{3}s_{23}s_{34}^{2}s_{45}^{4}$$

$$- 132s_{15}^{3}s_{23}s_{34}^{2}s_{45}^{4} + 1124\epsilon s_{12}s_{15}^{2}s_{23}s_{34}^{2}s_{45}^{4} + 317s_{12}s_{15}^{2}s_{23}s_{34}^{2}s_{45}^{4}$$

$$+ 2424\epsilon s_{12}^{2}s_{15}s_{23}s_{34}^{2}s_{45}^{4} + 510s_{12}^{2}s_{15}s_{23}s_{34}^{2}s_{45}^{4} - 252\epsilon s_{12}s_{15}^{4}s_{23}s_{45}^{4}$$

$$- 63s_{12}s_{15}^{4}s_{23}s_{45}^{4} + 292\epsilon s_{12}^{2}s_{15}^{3}s_{23}s_{45}^{4} + 73s_{12}^{2}s_{15}^{3}s_{23}s_{45}^{4} + 80\epsilon s_{12}^{3}s_{15}^{2}s_{23}s_{45}^{4}$$

$$+ 20s_{12}^{3}s_{15}^{2}s_{23}s_{45}^{4} + 16\epsilon s_{12}^{4}s_{15}s_{23}s_{45}^{4} + 4s_{12}^{4}s_{15}s_{23}s_{45}^{4} + 500\epsilon s_{12}s_{15}^{4}s_{34}s_{45}^{4}$$

$$+ 125s_{12}s_{15}^{4}s_{34}s_{45}^{4} - 36\epsilon s_{12}s_{23}^{4}s_{34}s_{45}^{4} - 9s_{12}s_{23}^{4}s_{34}s_{45}^{4} - 48\epsilon s_{15}s_{23}^{4}s_{34}s_{45}^{4}$$

$$- 12s_{15}s_{23}^{4}s_{34}s_{45}^{4} + 332\epsilon s_{12}^{2}s_{15}^{3}s_{34}s_{45}^{4} + 83s_{12}^{2}s_{15}^{3}s_{34}s_{45}^{4} - 324\epsilon s_{12}^{2}s_{23}^{3}s_{34}s_{45}^{4}$$

$$- 81s_{12}^{2}s_{23}^{3}s_{34}s_{45}^{4} - 176\epsilon s_{15}^{2}s_{23}^{3}s_{34}s_{45}^{4} - 32s_{15}^{2}s_{23}^{3}s_{34}s_{45}^{4} - 400\epsilon s_{12}s_{15}s_{23}^{3}s_{34}s_{45}^{4}$$

$$- 88s_{12}s_{15}s_{23}^{3}s_{34}s_{45}^{4} - 668\epsilon s_{12}^{3}s_{15}^{2}s_{34}s_{45}^{4} - 167s_{12}^{3}s_{15}^{2}s_{34}s_{45}^{4}$$

$$- 228\epsilon s_{12}^{3}s_{23}^{2}s_{34}s_{45}^{4} - 57s_{12}^{3}s_{23}^{2}s_{34}s_{45}^{4} + 64\epsilon s_{15}^{3}s_{23}^{2}s_{34}s_{45}^{4} + 4s_{15}^{3}s_{23}^{2}s_{34}s_{45}^{4}$$

$$- 1068\epsilon s_{12}s_{15}^{2}s_{23}^{2}s_{34}s_{45}^{4} - 219s_{12}s_{15}^{2}s_{23}^{2}s_{34}s_{45}^{4} - 664\epsilon s_{12}^{2}s_{15}s_{23}^{2}s_{34}s_{45}^{4}$$

$$- 106s_{12}^{2}s_{15}s_{23}^{2}s_{34}s_{45}^{4} - 4\epsilon s_{12}^{4}s_{15}s_{34}s_{45}^{4} - s_{12}^{4}s_{15}s_{34}s_{45}^{4} + 4\epsilon s_{12}^{4}s_{23}s_{34}s_{45}^{4}$$

$$+ s_{12}^{4}s_{23}s_{34}s_{45}^{4} + 160\epsilon s_{15}^{4}s_{23}s_{34}s_{45}^{4} + 40s_{15}^{4}s_{23}s_{34}s_{45}^{4} + 316\epsilon s_{12}s_{15}^{3}s_{23}s_{34}s_{45}^{4}$$

$$+ 43s_{12}s_{15}^{3}s_{23}s_{34}s_{45}^{4} - 1228\epsilon s_{12}^{2}s_{15}^{2}s_{23}s_{34}s_{45}^{4} - 307s_{12}^{2}s_{15}^{2}s_{23}s_{34}s_{45}^{4}$$

$$- 300\epsilon s_{12}^{3}s_{15}s_{23}s_{34}s_{45}^{4} - 39s_{12}^{3}s_{15}s_{23}s_{34}s_{45}^{4} + 120\epsilon s_{12}^{2}s_{34}^{5}s_{45}^{3} + 30s_{12}^{2}s_{34}^{5}s_{45}^{3}$$

$$+ 384\epsilon s_{23}^{2}s_{34}^{5}s_{45}^{3} + 72s_{23}^{2}s_{34}^{5}s_{45}^{3} - 184\epsilon s_{12}s_{15}s_{34}^{5}s_{45}^{3} - 46s_{12}s_{15}s_{34}^{5}s_{45}^{3}$$

$$+ 632\epsilon s_{12}s_{23}s_{34}^{5}s_{45}^{3} + 134s_{12}s_{23}s_{34}^{5}s_{45}^{3} - 192\epsilon s_{15}s_{23}s_{34}^{5}s_{45}^{3} - 36s_{15}s_{23}s_{34}^{5}s_{45}^{3}$$

$$+ 244\epsilon s_{12}^{3}s_{15}^{4}s_{45}^{3} + 61s_{12}^{3}s_{15}^{4}s_{45}^{3} + 4\epsilon s_{12}^{3}s_{23}^{4}s_{45}^{3} + s_{12}^{3}s_{23}^{4}s_{45}^{3}$$

$$+ 124\epsilon s_{12}^{2}s_{15}s_{23}^{4}s_{45}^{3} + 31s_{12}^{2}s_{15}s_{23}^{4}s_{45}^{3} - 120\epsilon s_{12}^{3}s_{34}^{4}s_{45}^{3} - 30s_{12}^{3}s_{34}^{4}s_{45}^{3}$$

$$- 656\epsilon s_{23}^{3}s_{34}^{4}s_{45}^{3} - 140s_{23}^{3}s_{34}^{4}s_{45}^{3} + 616\epsilon s_{12}s_{15}^{2}s_{34}^{4}s_{45}^{3} + 154s_{12}s_{15}^{2}s_{34}^{4}s_{45}^{3}$$

$$- 2256\epsilon s_{12}s_{23}^{2}s_{34}^{4}s_{45}^{3} - 444s_{12}s_{23}^{2}s_{34}^{4}s_{45}^{3} - 288\epsilon s_{15}s_{23}^{2}s_{34}^{4}s_{45}^{3} - 48s_{15}s_{23}^{2}s_{34}^{4}s_{45}^{3}$$

$$- 176\epsilon s_{12}^{2}s_{15}s_{34}^{4}s_{45}^{3} - 44s_{12}^{2}s_{15}s_{34}^{4}s_{45}^{3} - 1512\epsilon s_{12}^{2}s_{23}s_{34}^{4}s_{45}^{3} - 306s_{12}^{2}s_{23}s_{34}^{4}s_{45}^{3}$$

$$+ 464\epsilon s_{15}^{2}s_{23}s_{34}^{4}s_{45}^{3} + 92s_{15}^{2}s_{23}s_{34}^{4}s_{45}^{3} - 504\epsilon s_{12}s_{15}s_{23}s_{34}^{4}s_{45}^{3}$$

$$- 150s_{12}s_{15}s_{23}s_{34}^{4}s_{45}^{3} - 56\epsilon s_{12}^{4}s_{15}^{3}s_{45}^{3} - 14s_{12}^{4}s_{15}^{3}s_{45}^{3} + 52\epsilon s_{12}^{4}s_{23}^{3}s_{45}^{3}$$

$$+ 13s_{12}^{4}s_{23}^{3}s_{45}^{3} - 112\epsilon s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{45}^{3} - 28s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{45}^{3} + 360\epsilon s_{12}^{3}s_{15}s_{23}^{3}s_{45}^{3}$$

$$+ 90s_{12}^{3}s_{15}s_{23}^{3}s_{45}^{3} + 160\epsilon s_{23}^{4}s_{34}^{3}s_{45}^{3} + 40s_{23}^{4}s_{34}^{3}s_{45}^{3} - 680\epsilon s_{12}s_{15}^{3}s_{34}^{3}s_{45}^{3}$$

$$- 170s_{12}s_{15}^{3}s_{34}^{3}s_{45}^{3} + 1940\epsilon s_{12}s_{23}^{3}s_{34}^{3}s_{45}^{3} \quad + 437s_{12}s_{23}^{3}s_{34}^{3}s_{45}^{3}$$

$$+ 800\epsilon s_{15}s_{23}^{3}s_{34}^{3}s_{45}^{3} + 152s_{15}s_{23}^{3}s_{34}^{3}s_{45}^{3} - 688\epsilon s_{12}^{2}s_{15}^{2}s_{34}^{3}s_{45}^{3}$$

$$- 172s_{12}^{2}s_{15}^{2}s_{34}^{3}s_{45}^{3} + 3184\epsilon s_{12}^{2}s_{23}^{2}s_{34}^{3}s_{45}^{3} + 652s_{12}^{2}s_{23}^{2}s_{34}^{3}s_{45}^{3}$$

$$- 320\epsilon s_{15}^{2}s_{23}^{2}s_{34}^{3}s_{45}^{3} - 56s_{15}^{2}s_{23}^{2}s_{34}^{3}s_{45}^{3} + 2480\epsilon s_{12}s_{15}s_{23}^{2}s_{34}^{3}s_{45}^{3}$$

$$+ 452s_{12}s_{15}s_{23}^{2}s_{34}^{3}s_{45}^{3} + 544\epsilon s_{12}^{3}s_{15}s_{34}^{3}s_{45}^{3} + 136s_{12}^{3}s_{15}s_{34}^{3}s_{45}^{3}$$

$$+ 1032\epsilon s_{12}^{3}s_{23}s_{34}^{3}s_{45}^{3} + 198s_{12}^{3}s_{23}s_{34}^{3}s_{45}^{3} - 352\epsilon s_{15}^{3}s_{23}s_{34}^{3}s_{45}^{3}$$

$$- 76s_{15}^{3}s_{23}s_{34}^{3}s_{45}^{3} - 1072\epsilon s_{12}s_{15}^{2}s_{23}s_{34}^{3}s_{45}^{3} - 160s_{12}s_{15}^{2}s_{23}s_{34}^{3}s_{45}^{3}$$

$$+ 1808\epsilon s_{12}^{2}s_{15}s_{23}s_{34}^{3}s_{45}^{3} + 428s_{12}^{2}s_{15}s_{23}s_{34}^{3}s_{45}^{3} + 8\epsilon s_{12}^{5}s_{23}^{2}s_{45}^{3}$$

$$+ 2s_{12}^{5}s_{23}^{2}s_{45}^{3} - 276\epsilon s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{45}^{3} - 69s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{45}^{3} - 496\epsilon s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{45}^{3}$$

$$- 124s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{45}^{3} - 32\epsilon s_{12}^{4}s_{15}s_{23}^{2}s_{45}^{3} - 8s_{12}^{4}s_{15}s_{23}^{2}s_{45}^{3} + 248\epsilon s_{12}s_{15}^{4}s_{34}^{2}s_{45}^{3}$$

$$+ 62s_{12}s_{15}^{4}s_{34}^{2}s_{45}^{3} - 348\epsilon s_{12}s_{23}^{4}s_{34}^{2}s_{45}^{3} - 87s_{12}s_{23}^{4}s_{34}^{2}s_{45}^{3} - 160\epsilon s_{15}s_{23}^{4}s_{34}^{2}s_{45}^{3}$$

$$- 40s_{15}s_{23}^{4}s_{34}^{2}s_{45}^{3} + 1420\epsilon s_{12}^{2}s_{15}^{3}s_{34}^{2}s_{45}^{3} + 355s_{12}^{2}s_{15}^{3}s_{34}^{2}s_{45}^{3}$$

$$- 1688\epsilon s_{12}^{2}s_{23}^{3}s_{34}^{2}s_{45}^{3} - 398s_{12}^{2}s_{23}^{3}s_{34}^{2}s_{45}^{3} - 144\epsilon s_{15}^{2}s_{23}^{3}s_{34}^{2}s_{45}^{3}$$

$$- 12s_{15}^{2}s_{23}^{3}s_{34}^{2}s_{45}^{3} - 1336\epsilon s_{12}s_{15}s_{23}^{3}s_{34}^{2}s_{45}^{3} - 262s_{12}s_{15}s_{23}^{3}s_{34}^{2}s_{45}^{3}$$

$$- 348\epsilon s_{12}^{3}s_{15}^{2}s_{34}^{2}s_{45}^{3} - 87s_{12}^{3}s_{15}^{2}s_{34}^{2}s_{45}^{3} - 1536\epsilon s_{12}^{3}s_{23}^{2}s_{34}^{2}s_{45}^{3}$$

$$- 336s_{12}^{3}s_{23}^{2}s_{34}^{2}s_{45}^{3} + 224\epsilon s_{15}^{3}s_{23}^{2}s_{34}^{2}s_{45}^{3} + 32s_{15}^{3}s_{23}^{2}s_{34}^{2}s_{45}^{3}$$

$$- 808\epsilon s_{12}s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{3} - 178s_{12}s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{3} - 2064\epsilon s_{12}^{2}s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{3}$$

$$- 348s_{12}^{2}s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{3} - 124\epsilon s_{12}^{4}s_{15}s_{34}^{2}s_{45}^{3} - 31s_{12}^{4}s_{15}s_{34}^{2}s_{45}^{3}$$

$$
\begin{aligned}
&-212\epsilon s_{12}^4s_{23}s_{34}^2s_{45}^3-41s_{12}^4s_{23}s_{34}^2s_{45}^3+80\epsilon s_{15}^4s_{23}s_{34}^2s_{45}^3+20s_{15}^4s_{23}s_{34}^2s_{45}^3\\
&+1428\epsilon s_{12}s_{15}^3s_{23}s_{34}^2s_{45}^3+297s_{12}s_{15}^3s_{23}s_{34}^2s_{45}^3-588\epsilon s_{12}^2s_{15}^2s_{23}s_{34}^2s_{45}^3\\
&-231s_{12}^2s_{15}^2s_{23}s_{34}^2s_{45}^3-676\epsilon s_{12}^3s_{15}s_{23}s_{34}^2s_{45}^3-133s_{12}^3s_{15}s_{23}s_{34}^2s_{45}^3\\
&+264\epsilon s_{12}^2s_{15}^4s_{23}s_{45}^3+66s_{12}^2s_{15}^4s_{23}s_{45}^3-112\epsilon s_{12}^3s_{15}^3s_{23}s_{45}^3-28s_{12}^3s_{15}^3s_{23}s_{45}^3\\
&+36\epsilon s_{12}^4s_{15}^2s_{23}s_{45}^3+9s_{12}^4s_{15}^2s_{23}s_{45}^3-8\epsilon s_{12}^5s_{15}s_{23}s_{45}^3-2s_{12}^5s_{15}s_{23}s_{45}^3\\
&-676\epsilon s_{12}^2s_{15}^4s_{34}s_{45}^3-169s_{12}^2s_{15}^4s_{34}s_{45}^3+184\epsilon s_{12}^2s_{23}^4s_{34}s_{45}^3\\
&+46s_{12}^2s_{23}^4s_{34}s_{45}^3+68\epsilon s_{12}s_{15}s_{23}^4s_{34}s_{45}^3+17s_{12}s_{15}s_{23}^4s_{34}s_{45}^3\\
&-320\epsilon s_{12}^3s_{15}^3s_{34}s_{45}^3-80s_{12}^3s_{15}^3s_{34}s_{45}^3+412\epsilon s_{12}^3s_{23}^3s_{34}s_{45}^3+103s_{12}^3s_{23}^3s_{34}s_{45}^3\\
&+148\epsilon s_{12}s_{15}^2s_{23}^3s_{34}s_{45}^3+13s_{12}s_{15}^2s_{23}^3s_{34}s_{45}^3-116\epsilon s_{12}^2s_{15}s_{23}^3s_{34}s_{45}^3\\
&-53s_{12}^2s_{15}s_{23}^3s_{34}s_{45}^3+180\epsilon s_{12}^4s_{15}^2s_{34}s_{45}^3+45s_{12}^4s_{15}^2s_{34}s_{45}^3\\
&+216\epsilon s_{12}^4s_{23}^2s_{34}s_{45}^3+54s_{12}^4s_{23}^2s_{34}s_{45}^3+268\epsilon s_{12}s_{15}^3s_{23}^2s_{34}s_{45}^3\\
&+91s_{12}s_{15}^3s_{23}^2s_{34}s_{45}^3+1072\epsilon s_{12}^2s_{15}^2s_{23}^2s_{34}s_{45}^3+244s_{12}^2s_{15}^2s_{23}^2s_{34}s_{45}^3\\
&+72\epsilon s_{12}^3s_{15}s_{23}^2s_{34}s_{45}^3-30s_{12}^3s_{15}s_{23}^2s_{34}s_{45}^3-484\epsilon s_{12}s_{15}^4s_{23}s_{34}s_{45}^3\\
&-121s_{12}s_{15}^4s_{23}s_{34}s_{45}^3-160\epsilon s_{12}^2s_{15}^3s_{23}s_{34}s_{45}^3-4s_{12}^2s_{15}^3s_{23}s_{34}s_{45}^3\\
&+444\epsilon s_{12}^3s_{15}^2s_{23}s_{34}s_{45}^3+135s_{12}^3s_{15}^2s_{23}s_{34}s_{45}^3-92\epsilon s_{12}^4s_{15}s_{23}s_{34}s_{45}^3\\
&-35s_{12}^4s_{15}s_{23}s_{34}s_{45}^3-256\epsilon s_{23}^3s_{34}^5s_{45}^2-52s_{23}^3s_{34}^5s_{45}^2-792\epsilon s_{12}s_{23}^2s_{34}^5s_{45}^2\\
&-162s_{12}s_{23}^2s_{34}^5s_{45}^2+192\epsilon s_{15}s_{23}^2s_{34}^5s_{45}^2+36s_{15}s_{23}^2s_{34}^5s_{45}^2+120\epsilon s_{12}^2s_{15}s_{34}^5s_{45}^2\\
&+30s_{12}^2s_{15}s_{34}^5s_{45}^2-336\epsilon s_{12}^2s_{23}s_{34}^5s_{45}^2-72s_{12}^2s_{23}s_{34}^5s_{45}^2+448\epsilon s_{12}s_{15}s_{23}s_{34}^5s_{45}^2\\
&+88s_{12}s_{15}s_{23}s_{34}^5s_{45}^2-60\epsilon s_{12}^4s_{15}^4s_{45}^2-15s_{12}^4s_{15}^4s_{45}^2+8\epsilon s_{12}^4s_{23}^4s_{45}^2\\
&+2s_{12}^4s_{23}^4s_{45}^2-52\epsilon s_{12}^3s_{15}s_{23}^4s_{45}^2-13s_{12}^3s_{15}s_{23}^4s_{45}^2+176\epsilon s_{23}^4s_{34}^4s_{45}^2\\
&+44s_{23}^4s_{34}^4s_{45}^2+1520\epsilon s_{12}s_{23}^3s_{34}^4s_{45}^2+332s_{12}s_{23}^3s_{34}^4s_{45}^2+208\epsilon s_{15}s_{23}^3s_{34}^4s_{45}^2\\
&+28s_{15}s_{23}^3s_{34}^4s_{45}^2-544\epsilon s_{12}^2s_{15}^2s_{34}^4s_{45}^2-136s_{12}^2s_{15}^2s_{34}^4s_{45}^2\\
&+2144\epsilon s_{12}^2s_{23}^2s_{34}^4s_{45}^2+440s_{12}^2s_{23}^2s_{34}^4s_{45}^2-320\epsilon s_{15}^2s_{23}^2s_{34}^4s_{45}^2\\
&-56s_{15}^2s_{23}^2s_{34}^4s_{45}^2+400\epsilon s_{12}s_{15}s_{23}^2s_{34}^4s_{45}^2+76s_{12}s_{15}s_{23}^2s_{34}^4s_{45}^2\\
&+552\epsilon s_{12}^3s_{23}s_{34}^4s_{45}^2+114s_{12}^3s_{23}s_{34}^4s_{45}^2-1064\epsilon s_{12}s_{15}^2s_{23}s_{34}^4s_{45}^2\\
&-218s_{12}s_{15}^2s_{23}s_{34}^4s_{45}^2-192\epsilon s_{12}^2s_{15}s_{23}s_{34}^4s_{45}^2-24\epsilon s_{12}^5s_{23}^3s_{45}^2-6s_{12}^5s_{23}^3s_{45}^2\\
&+12\epsilon s_{12}^3s_{15}^2s_{23}^3s_{45}^2+3s_{12}^3s_{15}^2s_{23}^3s_{45}^2-132\epsilon s_{12}^4s_{15}s_{23}^3s_{45}^2-33s_{12}^4s_{15}s_{23}^3s_{45}^2\\
&-624\epsilon s_{12}s_{23}^4s_{34}^3s_{45}^2-156s_{12}s_{23}^4s_{34}^3s_{45}^2-176\epsilon s_{15}s_{23}^4s_{34}^3s_{45}^2-44s_{15}s_{23}^4s_{34}^3s_{45}^2\\
&+728\epsilon s_{12}^2s_{15}^3s_{34}^3s_{45}^2+182s_{12}^2s_{15}^3s_{34}^3s_{45}^2-2500\epsilon s_{12}^2s_{23}^3s_{34}^3s_{45}^2\\
&-577s_{12}^2s_{23}^3s_{34}^3s_{45}^2+48\epsilon s_{15}^2s_{23}^3s_{34}^3s_{45}^2+24s_{15}^2s_{23}^3s_{34}^3s_{45}^2\\
&-1028\epsilon s_{12}s_{15}s_{23}^3s_{34}^3s_{45}^2-173s_{12}s_{15}s_{23}^3s_{34}^3s_{45}^2+428\epsilon s_{12}^3s_{15}^2s_{34}^3s_{45}^2\\
&+107s_{12}^3s_{15}^2s_{34}^3s_{45}^2-1980\epsilon s_{12}^3s_{23}^2s_{34}^3s_{45}^2-423s_{12}^3s_{23}^2s_{34}^3s_{45}^2\\
&+128\epsilon s_{15}^3s_{23}^2s_{34}^3s_{45}^2+20s_{15}^3s_{23}^2s_{34}^3s_{45}^2+356\epsilon s_{12}s_{15}^2s_{23}^2s_{34}^3s_{45}^2\\
&+41s_{12}s_{15}^2s_{23}^2s_{34}^3s_{45}^2-748\epsilon s_{12}^2s_{15}s_{23}^2s_{34}^3s_{45}^2-79s_{12}^2s_{15}s_{23}^2s_{34}^3s_{45}^2\\
&-120\epsilon s_{12}^4s_{15}s_{34}^3s_{45}^2-30s_{12}^4s_{15}s_{34}^3s_{45}^2-216\epsilon s_{12}^4s_{23}s_{34}^3s_{45}^2-42s_{12}^4s_{23}s_{34}^3s_{45}^2\\
&+848\epsilon s_{12}s_{15}^3s_{23}s_{34}^3s_{45}^2+188s_{12}s_{15}^3s_{23}s_{34}^3s_{45}^2+876\epsilon s_{12}^2s_{15}^2s_{23}s_{34}^3s_{45}^2\\
&+111s_{12}^2s_{15}^2s_{23}s_{34}^3s_{45}^2+20\epsilon s_{12}^3s_{15}s_{23}s_{34}^3s_{45}^2-19s_{12}^3s_{15}s_{23}s_{34}^3s_{45}^2\\
&+132\epsilon s_{12}^3s_{15}^3s_{23}^2s_{45}^2+33s_{12}^3s_{15}^3s_{23}^2s_{45}^2+180\epsilon s_{12}^4s_{15}^2s_{23}^2s_{45}^2\\
&+45s_{12}^4s_{15}^2s_{23}^2s_{45}^2+48\epsilon s_{12}^5s_{15}s_{23}^2s_{45}^2+12s_{12}^5s_{15}s_{23}^2s_{45}^2-304\epsilon s_{12}^2s_{15}^4s_{34}^2s_{45}^2\\
&-76s_{12}^2s_{15}^4s_{34}^2s_{45}^2+668\epsilon s_{12}^2s_{23}^4s_{34}^2s_{45}^2+167s_{12}^2s_{23}^4s_{34}^2s_{45}^2\\
&+388\epsilon s_{12}s_{15}s_{23}^4s_{34}^2s_{45}^2+97s_{12}s_{15}s_{23}^4s_{34}^2s_{45}^2-792\epsilon s_{12}^3s_{15}^3s_{34}^2s_{45}^2\\
&-198s_{12}^3s_{15}^3s_{34}^2s_{45}^2+1536\epsilon s_{12}^3s_{23}^3s_{34}^2s_{45}^2+372s_{12}^3s_{23}^3s_{34}^2s_{45}^2\\
&-128\epsilon s_{12}s_{15}^2s_{23}^3s_{34}^2s_{45}^2-68s_{12}s_{15}^2s_{23}^3s_{34}^2s_{45}^2+528\epsilon s_{12}^2s_{15}s_{23}^3s_{34}^2s_{45}^2\\
&+72s_{12}^2s_{15}s_{23}^3s_{34}^2s_{45}^2+56\epsilon s_{12}^4s_{15}^2s_{34}^2s_{45}^2+14s_{12}^4s_{15}^2s_{34}^2s_{45}^2\\
&+732\epsilon s_{12}^4s_{23}^2s_{34}^2s_{45}^2+171s_{12}^4s_{23}^2s_{34}^2s_{45}^2-28\epsilon s_{12}s_{15}^3s_{23}^2s_{34}^2s_{45}^2
\end{aligned}
$$

$$
\begin{aligned}
&+29s_{12}s_{15}^{3}s_{23}^{2}s_{34}^{2}s_{45}^{2}+224\epsilon s_{12}^{2}s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{2}+56s_{12}^{2}s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}^{2}\\
&-240\epsilon s_{12}^{3}s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{2}-132s_{12}^{3}s_{15}s_{23}^{2}s_{34}^{2}s_{45}^{2}-232\epsilon s_{12}s_{15}^{4}s_{23}s_{34}^{2}s_{45}^{2}\\
&-58s_{12}s_{15}^{4}s_{23}s_{34}^{2}s_{45}^{2}-828\epsilon s_{12}^{2}s_{15}^{3}s_{23}s_{34}^{2}s_{45}^{2}-159s_{12}^{2}s_{15}^{3}s_{23}s_{34}^{2}s_{45}^{2}\\
&+192\epsilon s_{12}^{3}s_{15}^{2}s_{23}s_{34}^{2}s_{45}^{2}+108s_{12}^{3}s_{15}^{2}s_{23}s_{34}^{2}s_{45}^{2}-380\epsilon s_{12}^{4}s_{15}s_{23}s_{34}^{2}s_{45}^{2}\\
&-95s_{12}^{4}s_{15}s_{23}s_{34}^{2}s_{45}^{2}-92\epsilon s_{12}^{3}s_{15}^{4}s_{23}s_{45}^{2}-23s_{12}^{3}s_{15}^{4}s_{23}s_{45}^{2}+4\epsilon s_{12}^{4}s_{15}^{3}s_{23}s_{45}^{2}\\
&+s_{12}^{4}s_{15}^{3}s_{23}s_{45}^{2}-24\epsilon s_{12}^{5}s_{15}^{2}s_{23}s_{45}^{2}-6s_{12}^{5}s_{15}^{2}s_{23}s_{45}^{2}+364\epsilon s_{12}^{3}s_{15}^{4}s_{34}s_{45}^{2}\\
&+91s_{12}^{3}s_{15}^{4}s_{34}s_{45}^{2}-228\epsilon s_{12}^{3}s_{23}^{4}s_{34}s_{45}^{2}-57s_{12}^{3}s_{23}^{4}s_{34}s_{45}^{2}-40\epsilon s_{12}^{2}s_{15}s_{23}^{4}s_{34}s_{45}^{2}\\
&-10s_{12}^{2}s_{15}s_{23}^{4}s_{34}s_{45}^{2}+124\epsilon s_{12}^{4}s_{15}^{3}s_{34}s_{45}^{2}+31s_{12}^{4}s_{15}^{3}s_{34}s_{45}^{2}\\
&-276\epsilon s_{12}^{4}s_{23}^{3}s_{34}s_{45}^{2}-69s_{12}^{4}s_{23}^{3}s_{34}s_{45}^{2}+160\epsilon s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{34}s_{45}^{2}\\
&+52s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{34}s_{45}^{2}+448\epsilon s_{12}^{3}s_{15}s_{23}^{3}s_{34}s_{45}^{2}+124s_{12}^{3}s_{15}s_{23}^{3}s_{34}s_{45}^{2}\\
&-104\epsilon s_{12}^{5}s_{23}^{2}s_{34}s_{45}^{2}-26s_{12}^{5}s_{23}^{2}s_{34}s_{45}^{2}-536\epsilon s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{34}s_{45}^{2}\\
&-146s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{34}s_{45}^{2}-504\epsilon s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{34}s_{45}^{2}-126s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{34}s_{45}^{2}\\
&+252\epsilon s_{12}^{4}s_{15}s_{23}^{2}s_{34}s_{45}^{2}+75s_{12}^{4}s_{15}s_{23}^{2}s_{34}s_{45}^{2}+416\epsilon s_{12}^{2}s_{15}^{4}s_{23}s_{34}s_{45}^{2}\\
&+104s_{12}^{2}s_{15}^{4}s_{23}s_{34}s_{45}^{2}-80\epsilon s_{12}^{3}s_{15}^{3}s_{23}s_{34}s_{45}^{2}-32s_{12}^{3}s_{15}^{3}s_{23}s_{34}s_{45}^{2}\\
&-100\epsilon s_{12}^{4}s_{15}^{2}s_{23}s_{34}s_{45}^{2}-37s_{12}^{4}s_{15}^{2}s_{23}s_{34}s_{45}^{2}+104\epsilon s_{12}^{5}s_{15}s_{23}s_{34}s_{45}^{2}\\
&+26s_{12}^{5}s_{15}s_{23}s_{34}s_{45}^{2}+64\epsilon s_{23}^{4}s_{34}^{5}s_{45}+16s_{23}^{4}s_{34}^{5}s_{45}+440\epsilon s_{12}s_{23}^{3}s_{34}^{5}s_{45}\\
&+98s_{12}s_{23}^{3}s_{34}^{5}s_{45}-64\epsilon s_{15}s_{23}^{3}s_{34}^{5}s_{45}-16s_{15}s_{23}^{3}s_{34}^{5}s_{45}+312\epsilon s_{12}^{2}s_{23}^{2}s_{34}^{5}s_{45}\\
&+66s_{12}^{2}s_{23}^{2}s_{34}^{5}s_{45}-344\epsilon s_{12}s_{15}s_{23}^{2}s_{34}^{5}s_{45}-74s_{12}s_{15}s_{23}^{2}s_{34}^{5}s_{45}\\
&-216\epsilon s_{12}^{2}s_{15}s_{23}s_{34}^{5}s_{45}-42s_{12}^{2}s_{15}s_{23}s_{34}^{5}s_{45}-408\epsilon s_{12}s_{23}^{4}s_{34}^{4}s_{45}\\
&-102s_{12}s_{23}^{4}s_{34}^{4}s_{45}-64\epsilon s_{15}s_{23}^{4}s_{34}^{4}s_{45}-16s_{15}s_{23}^{4}s_{34}^{4}s_{45}-1288\epsilon s_{12}^{2}s_{23}^{3}s_{34}^{4}s_{45}\\
&-298s_{12}^{2}s_{23}^{3}s_{34}^{4}s_{45}+64\epsilon s_{15}^{2}s_{23}^{3}s_{34}^{4}s_{45}+16s_{15}^{2}s_{23}^{3}s_{34}^{4}s_{45}-152\epsilon s_{12}s_{15}s_{23}^{3}s_{34}^{4}s_{45}\\
&-14s_{12}s_{15}s_{23}^{3}s_{34}^{4}s_{45}+120\epsilon s_{12}^{3}s_{15}^{2}s_{34}^{4}s_{45}+30s_{12}^{3}s_{15}^{2}s_{34}^{4}s_{45}-720\epsilon s_{12}^{3}s_{23}^{2}s_{34}^{4}s_{45}\\
&-156s_{12}^{3}s_{23}^{2}s_{34}^{4}s_{45}+464\epsilon s_{12}s_{15}^{2}s_{23}^{2}s_{34}^{4}s_{45}+92s_{12}s_{15}^{2}s_{23}^{2}s_{34}^{4}s_{45}\\
&+576\epsilon s_{12}^{2}s_{15}s_{23}^{2}s_{34}^{4}s_{45}+144s_{12}^{2}s_{15}s_{23}^{2}s_{34}^{4}s_{45}+424\epsilon s_{12}^{2}s_{15}^{2}s_{23}s_{34}^{4}s_{45}\\
&+82s_{12}^{2}s_{15}^{2}s_{23}s_{34}^{4}s_{45}+408\epsilon s_{12}^{3}s_{15}s_{23}s_{34}^{4}s_{45}+78s_{12}^{3}s_{15}s_{23}s_{34}^{4}s_{45}\\
&+744\epsilon s_{12}^{2}s_{23}^{4}s_{34}^{3}s_{45}+186s_{12}^{2}s_{23}^{4}s_{34}^{3}s_{45}+344\epsilon s_{12}s_{15}s_{23}^{4}s_{34}^{3}s_{45}\\
&+86s_{12}s_{15}s_{23}^{4}s_{34}^{3}s_{45}-240\epsilon s_{12}^{3}s_{15}^{3}s_{34}^{3}s_{45}-60s_{12}^{3}s_{15}^{3}s_{34}^{3}s_{45}\\
&+1344\epsilon s_{12}^{3}s_{23}^{3}s_{34}^{3}s_{45}+324s_{12}^{3}s_{23}^{3}s_{34}^{3}s_{45}-224\epsilon s_{12}s_{15}^{2}s_{23}^{3}s_{34}^{3}s_{45}\\
&-68s_{12}s_{15}^{2}s_{23}^{3}s_{34}^{3}s_{45}+36\epsilon s_{12}^{2}s_{15}s_{23}^{3}s_{34}^{3}s_{45}-27s_{12}^{2}s_{15}s_{23}^{3}s_{34}^{3}s_{45}\\
&-120\epsilon s_{12}^{4}s_{15}^{2}s_{34}^{3}s_{45}-30s_{12}^{4}s_{15}^{2}s_{34}^{3}s_{45}+504\epsilon s_{12}^{4}s_{23}^{2}s_{34}^{3}s_{45}+114s_{12}^{4}s_{23}^{2}s_{34}^{3}s_{45}\\
&-120\epsilon s_{12}s_{15}^{3}s_{23}^{2}s_{34}^{3}s_{45}-18s_{12}s_{15}^{3}s_{23}^{2}s_{34}^{3}s_{45}-324\epsilon s_{12}^{2}s_{15}^{2}s_{23}^{2}s_{34}^{3}s_{45}\\
&-57s_{12}^{2}s_{15}^{2}s_{23}^{2}s_{34}^{3}s_{45}-788\epsilon s_{12}^{3}s_{15}s_{23}^{2}s_{34}^{3}s_{45}-221s_{12}^{3}s_{15}s_{23}^{2}s_{34}^{3}s_{45}\\
&-360\epsilon s_{12}^{2}s_{15}^{3}s_{23}s_{34}^{3}s_{45}-78s_{12}^{2}s_{15}^{3}s_{23}s_{34}^{3}s_{45}-124\epsilon s_{12}^{3}s_{15}^{2}s_{23}s_{34}^{3}s_{45}\\
&+5s_{12}^{3}s_{15}^{2}s_{23}s_{34}^{3}s_{45}-288\epsilon s_{12}^{4}s_{15}s_{23}s_{34}^{3}s_{45}-60s_{12}^{4}s_{15}s_{23}s_{34}^{3}s_{45}\\
&+120\epsilon s_{12}^{3}s_{15}^{4}s_{34}^{2}s_{45}+30s_{12}^{3}s_{15}^{4}s_{34}^{2}s_{45}-520\epsilon s_{12}^{3}s_{23}^{4}s_{34}^{2}s_{45}\\
&-130s_{12}^{3}s_{23}^{4}s_{34}^{2}s_{45}-340\epsilon s_{12}^{2}s_{15}s_{23}^{4}s_{34}^{2}s_{45}-85s_{12}^{2}s_{15}s_{23}^{4}s_{34}^{2}s_{45}\\
&+180\epsilon s_{12}^{4}s_{15}^{3}s_{34}^{2}s_{45}+45s_{12}^{4}s_{15}^{3}s_{34}^{2}s_{45}-584\epsilon s_{12}^{4}s_{23}^{3}s_{34}^{2}s_{45}-146s_{12}^{4}s_{23}^{3}s_{34}^{2}s_{45}\\
&+528\epsilon s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{34}^{2}s_{45}+144s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{34}^{2}s_{45}+424\epsilon s_{12}^{3}s_{15}s_{23}^{3}s_{34}^{2}s_{45}\\
&+118s_{12}^{3}s_{15}s_{23}^{3}s_{34}^{2}s_{45}-96\epsilon s_{12}^{5}s_{23}^{2}s_{34}^{2}s_{45}-24s_{12}^{5}s_{23}^{2}s_{34}^{2}s_{45}\\
&-340\epsilon s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{34}^{2}s_{45}-97s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{34}^{2}s_{45}+8\epsilon s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}\\
&+2s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{34}^{2}s_{45}+732\epsilon s_{12}^{4}s_{15}s_{23}^{2}s_{34}^{2}s_{45}+195s_{12}^{4}s_{15}s_{23}^{2}s_{34}^{2}s_{45}\\
&+152\epsilon s_{12}^{2}s_{15}^{4}s_{23}s_{34}^{2}s_{45}+38s_{12}^{2}s_{15}^{4}s_{23}s_{34}^{2}s_{45}-32\epsilon s_{12}^{3}s_{15}^{3}s_{23}s_{34}^{2}s_{45}\\
&-20s_{12}^{3}s_{15}^{3}s_{23}s_{34}^{2}s_{45}-328\epsilon s_{12}^{4}s_{15}^{2}s_{23}s_{34}^{2}s_{45}-94s_{12}^{4}s_{15}^{2}s_{23}s_{34}^{2}s_{45}\\
&+96\epsilon s_{12}^{5}s_{15}s_{23}s_{34}^{2}s_{45}+24s_{12}^{5}s_{15}s_{23}s_{34}^{2}s_{45}-60\epsilon s_{12}^{4}s_{15}^{4}s_{34}s_{45}-15s_{12}^{4}s_{15}^{4}s_{34}s_{45}\\
&+120\epsilon s_{12}^{4}s_{23}^{4}s_{34}s_{45}+30s_{12}^{4}s_{23}^{4}s_{34}s_{45}+60\epsilon s_{12}^{3}s_{15}s_{23}^{4}s_{34}s_{45}+15s_{12}^{3}s_{15}s_{23}^{4}s_{34}s_{45}
\end{aligned}
$$

$$
\begin{aligned}
&+ 88\epsilon s_{12}^{5}s_{23}^{3}s_{34}s_{45} + 22s_{12}^{5}s_{23}^{3}s_{34}s_{45} - 212\epsilon s_{12}^{3}s_{15}^{2}s_{23}^{3}s_{34}s_{45}\\
&- 53s_{12}^{3}s_{15}^{2}s_{23}^{3}s_{34}s_{45} - 244\epsilon s_{12}^{4}s_{15}s_{23}^{3}s_{34}s_{45} - 61s_{12}^{4}s_{15}s_{23}^{3}s_{34}s_{45}\\
&+ 244\epsilon s_{12}^{3}s_{15}^{3}s_{23}^{2}s_{34}s_{45} + 61s_{12}^{3}s_{15}^{3}s_{23}^{2}s_{34}s_{45} + 68\epsilon s_{12}^{4}s_{15}^{2}s_{23}^{2}s_{34}s_{45}\\
&+ 17s_{12}^{4}s_{15}^{2}s_{23}^{2}s_{34}s_{45} - 176\epsilon s_{12}^{5}s_{15}s_{23}^{2}s_{34}s_{45} - 44s_{12}^{5}s_{15}s_{23}^{2}s_{34}s_{45}\\
&- 92\epsilon s_{12}^{3}s_{15}^{4}s_{23}s_{34}s_{45} - 23s_{12}^{3}s_{15}^{4}s_{23}s_{34}s_{45} + 116\epsilon s_{12}^{4}s_{15}^{3}s_{23}s_{34}s_{45}\\
&+ 29s_{12}^{4}s_{15}^{3}s_{23}s_{34}s_{45} + 88\epsilon s_{12}^{5}s_{15}^{2}s_{23}s_{34}s_{45} + 22s_{12}^{5}s_{15}^{2}s_{23}s_{34}s_{45} - 96\epsilon s_{12}s_{23}^{4}s_{34}^{5}\\
&- 24s_{12}s_{23}^{4}s_{34}^{5} - 96\epsilon s_{12}^{2}s_{23}^{3}s_{34}^{5} - 24s_{12}^{2}s_{23}^{3}s_{34}^{5} + 96\epsilon s_{12}s_{15}s_{23}^{3}s_{34}^{5}\\
&+ 24s_{12}s_{15}s_{23}^{3}s_{34}^{5} + 96\epsilon s_{12}^{2}s_{15}s_{23}^{2}s_{34}^{5} + 24s_{12}^{2}s_{15}s_{23}^{2}s_{34}^{5} + 288\epsilon s_{12}^{2}s_{23}^{4}s_{34}^{4}\\
&+ 72s_{12}^{2}s_{23}^{4}s_{34}^{4} + 96\epsilon s_{12}s_{15}s_{23}^{4}s_{34}^{4} + 24s_{12}s_{15}s_{23}^{4}s_{34}^{4} + 288\epsilon s_{12}^{3}s_{23}^{3}s_{34}^{4}\\
&+ 72s_{12}^{3}s_{23}^{3}s_{34}^{4} - 96\epsilon s_{12}s_{15}^{2}s_{23}^{3}s_{34}^{4} - 24s_{12}s_{15}^{2}s_{23}^{3}s_{34}^{4} - 288\epsilon s_{12}^{2}s_{15}s_{23}^{3}s_{34}^{4}\\
&- 72s_{12}^{2}s_{15}s_{23}^{3}s_{34}^{4} - 384\epsilon s_{12}^{3}s_{15}s_{23}^{2}s_{34}^{4} - 96s_{12}^{3}s_{15}s_{23}^{2}s_{34}^{4} + 96\epsilon s_{12}^{3}s_{15}^{2}s_{23}s_{34}^{4}\\
&+ 24s_{12}^{3}s_{15}^{2}s_{23}s_{34}^{4} - 288\epsilon s_{12}^{3}s_{23}^{4}s_{34}^{3} - 72s_{12}^{3}s_{23}^{4}s_{34}^{3} - 192\epsilon s_{12}^{2}s_{15}s_{23}^{4}s_{34}^{3}\\
&- 48s_{12}^{2}s_{15}s_{23}^{4}s_{34}^{3} - 288\epsilon s_{12}^{4}s_{23}^{3}s_{34}^{3} - 72s_{12}^{4}s_{23}^{3}s_{34}^{3} + 288\epsilon s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{34}^{3}\\
&+ 72s_{12}^{2}s_{15}^{2}s_{23}^{3}s_{34}^{3} + 288\epsilon s_{12}^{3}s_{15}s_{23}^{3}s_{34}^{3} + 72s_{12}^{3}s_{15}s_{23}^{3}s_{34}^{3} - 96\epsilon s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{34}^{3}\\
&- 24s_{12}^{2}s_{15}^{3}s_{23}^{2}s_{34}^{3} + 96\epsilon s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{34}^{3} + 24s_{12}^{3}s_{15}^{2}s_{23}^{2}s_{34}^{3}\\
&+ 480\epsilon s_{12}^{4}s_{15}s_{23}^{2}s_{34}^{3} + 120s_{12}^{4}s_{15}s_{23}^{2}s_{34}^{3} - 96\epsilon s_{12}^{3}s_{15}^{3}s_{23}s_{34}^{3} - 24s_{12}^{3}s_{15}^{3}s_{23}s_{34}^{3}\\
&- 192\epsilon s_{12}^{4}s_{15}^{2}s_{23}s_{34}^{3} - 48s_{12}^{4}s_{15}^{2}s_{23}s_{34}^{3} + 96\epsilon s_{12}^{4}s_{23}^{4}s_{34}^{2} + 24s_{12}^{4}s_{23}^{4}s_{34}^{2}\\
&+ 96\epsilon s_{12}^{3}s_{15}s_{23}^{4}s_{34}^{2} + 24s_{12}^{3}s_{15}s_{23}^{4}s_{34}^{2} + 96\epsilon s_{12}^{5}s_{23}^{3}s_{34}^{2} + 24s_{12}^{5}s_{23}^{3}s_{34}^{2}\\
&- 192\epsilon s_{12}^{3}s_{15}^{2}s_{23}^{3}s_{34}^{2} - 48s_{12}^{3}s_{15}^{2}s_{23}^{3}s_{34}^{2} - 96\epsilon s_{12}^{4}s_{15}s_{23}^{3}s_{34}^{2} - 24s_{12}^{4}s_{15}s_{23}^{3}s_{34}^{2}\\
&+ 96\epsilon s_{12}^{3}s_{15}^{3}s_{23}^{2}s_{34}^{2} + 24s_{12}^{3}s_{15}^{3}s_{23}^{2}s_{34}^{2} - 96\epsilon s_{12}^{4}s_{15}^{2}s_{23}^{2}s_{34}^{2} - 24s_{12}^{4}s_{15}^{2}s_{23}^{2}s_{34}^{2}\\
&- 192\epsilon s_{12}^{5}s_{15}s_{23}^{2}s_{34}^{2} - 48s_{12}^{5}s_{15}s_{23}^{2}s_{34}^{2} + 96\epsilon s_{12}^{4}s_{15}^{3}s_{23}s_{34}^{2} + 24s_{12}^{4}s_{15}^{3}s_{23}s_{34}^{2}\\
&+ 96\epsilon s_{12}^{5}s_{15}^{2}s_{23}s_{34}^{2} + 24s_{12}^{5}s_{15}^{2}s_{23}s_{34}^{2})/(8(4\epsilon+1)s_{12}s_{23}s_{45}^{2}(s_{12}-s_{45})(s_{34}+s_{45})\\
&(s_{12}+s_{15}-s_{34})(s_{12}+s_{23}-s_{45})(s_{12}-s_{34}-s_{45})(-s_{15}+s_{23}+s_{34})(-s_{15}+s_{23}-s_{45}))
\end{aligned}
\tag{B.2}
$$

After running our algorithm, c_{107} is reduced to about 9 lines long:

$$
\begin{aligned}
c_{107} = &\frac{3s_{23}s_{34}}{2(4\epsilon+1)s_{12}s_{45}(-s_{15}+s_{23}+s_{34})} - \frac{3s_{34}}{2(4\epsilon+1)s_{12}s_{45}} + \frac{15s_{15}^{2}-15s_{15}s_{34}}{8s_{23}s_{45}(-s_{12}-s_{15}+s_{34})}\\
&+ \frac{s_{23}s_{34}^{2}}{s_{45}^{2}(s_{45}-s_{12})(-s_{15}+s_{23}+s_{34})} - \frac{2s_{23}s_{34}^{2}}{s_{12}s_{45}^{2}(-s_{15}+s_{23}+s_{34})}\\
&+ \frac{s_{23}s_{34}+s_{34}^{2}}{s_{45}(s_{45}-s_{12})(-s_{15}+s_{23}+s_{34})} - \frac{11s_{23}s_{34}}{2s_{12}s_{45}(-s_{15}+s_{23}+s_{34})}\\
&- \frac{15s_{15}s_{34}}{8s_{23}s_{45}(-s_{12}+s_{34}+s_{45})} + \frac{s_{15}-s_{23}-s_{34}}{s_{45}(-s_{12}-s_{23}+s_{45})} + \frac{2s_{15}-2s_{34}}{s_{12}s_{23}}\\
&+ \frac{15s_{15}-15s_{34}}{8s_{23}(-s_{12}-s_{15}+s_{34})} - \frac{7s_{23}}{2s_{12}(-s_{15}+s_{23}+s_{34})}\\
&- \frac{15s_{23}}{4(-s_{12}-s_{15}+s_{34})(-s_{15}+s_{23}+s_{34})} - \frac{s_{15}}{2s_{12}(-s_{12}-s_{23}+s_{45})}\\
&+ \frac{s_{23}-s_{45}}{2s_{12}(s_{15}-s_{23}+s_{45})} + \frac{15s_{15}}{8s_{45}(-s_{12}-s_{15}+s_{34})} + \frac{15}{4(-s_{12}-s_{15}+s_{34})}\\
&+ \frac{7s_{34}}{4s_{23}(-s_{12}-s_{23}+s_{45})} - \frac{5s_{34}}{4(s_{45}-s_{12})(-s_{12}-s_{23}+s_{45})} - \frac{15s_{34}}{8s_{23}(-s_{12}+s_{34}+s_{45})}\\
&+ \frac{1}{2(-s_{12}-s_{23}+s_{45})} + \frac{4s_{34}}{s_{12}s_{45}} - \frac{11s_{34}}{4s_{45}(s_{45}-s_{12})} - \frac{15}{8(-s_{12}+s_{34}+s_{45})}
\end{aligned}
$$

$$+\frac{5}{4(s_{45}-s_{12})}+\frac{4}{s_{12}}-\frac{3{s_{34}}^2}{{s_{45}}^2(-s_{15}+s_{23}+s_{34})}+\frac{s_{34}}{4s_{45}(-s_{15}+s_{23}+s_{34})}$$
$$+\frac{s_{45}}{2(s_{34}+s_{45})(s_{15}-s_{23}+s_{45})}+\frac{3s_{34}}{{s_{45}}^2}-\frac{1}{2(s_{34}+s_{45})}-\frac{1}{4s_{45}} \tag{B.3}$$

For this particular coefficient, we achieve a byte size reduction of a factor from 249464 bytes to 15192 bytes, which is 6% of the original size. Besides the reduction of the byte-size, we see that the highest total degree of the numerator of c_{107} also significantly decreased from 11 to 3.

References

[1] G. Apollinari et al. eds., *High-Luminosity Large Hadron Collider (HL-LHC)*, Technical Design Report V. 0.1. CERN-2017-007-M (2017).

[2] FCC collaboration, *HE-LHC: The High-Energy Large Hadron Collider*: Future Circular Collider Conceptual Design Report Volume 4, *Eur. Phys. J. ST* **228** (2019) 1109 [INSPIRE].

[3] F.V. Tkachov, *A Theorem on Analytical Calculability of Four Loop Renormalization Group Functions*, *Phys. Lett. B* **100** (1981) 65 [INSPIRE].

[4] K.G. Chetyrkin and F.V. Tkachov, *Integration by Parts: The Algorithm to Calculate β-functions in 4 Loops*, *Nucl. Phys. B* **192** (1981) 159 [INSPIRE].

[5] C. Anastasiou and A. Lazopoulos, *Automatic integral reduction for higher order perturbative calculations*, *JHEP* **07** (2004) 046 [hep-ph/0404258] [INSPIRE].

[6] A.V. Smirnov, *Algorithm FIRE – Feynman Integral REduction*, *JHEP* **10** (2008) 107 [arXiv:0807.3243] [INSPIRE].

[7] A.V. Smirnov and V.A. Smirnov, *FIRE4, LiteRed and accompanying tools to solve integration by parts relations*, *Comput. Phys. Commun.* **184** (2013) 2820 [arXiv:1302.5885] [INSPIRE].

[8] A.V. Smirnov, *FIRE5: a C++ implementation of Feynman Integral REduction*, *Comput. Phys. Commun.* **189** (2015) 182 [arXiv:1408.2372] [INSPIRE].

[9] A.V. Smirnov and F.S. Chuharev, *FIRE6: Feynman Integral REduction with Modular Arithmetic*, arXiv:1901.07808 [INSPIRE].

[10] P. Maierhöfer, J. Usovitsch and P. Uwer, *Kira — A Feynman integral reduction program*, *Comput. Phys. Commun.* **230** (2018) 99 [arXiv:1705.05610] [INSPIRE].

[11] P. Maierhöfer and J. Usovitsch, *Kira 1.2 Release Notes*, arXiv:1812.01491 [INSPIRE].

[12] P. Maierhoefer and J. Usovitsch, *Recent developments in Kira*, *CERN Yellow Rep. Monogr.* **3** (2020) 201.

[13] C. Studerus, *Reduze-Feynman Integral Reduction in C++*, *Comput. Phys. Commun.* **181** (2010) 1293 [arXiv:0912.2546] [INSPIRE].

[14] A. von Manteuffel and C. Studerus, *Reduze 2 — Distributed Feynman Integral Reduction*, arXiv:1201.4330 [INSPIRE].

[15] R.N. Lee, *LiteRed 1.4: a powerful tool for reduction of multiloop integrals*, *J. Phys. Conf. Ser.* **523** (2014) 012059 [arXiv:1310.1145] [INSPIRE].

[16] J. Klappert, F. Lange, P. Maierhöfer and J. Usovitsch, *Integral Reduction with Kira 2.0 and Finite Field Methods*, arXiv:2008.06494 [INSPIRE].

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

[18] A.V. Smirnov and V.A. Smirnov, *S-bases as a tool to solve reduction problems for Feynman integrals*, *Nucl. Phys. B Proc. Suppl.* **160** (2006) 80 [hep-ph/0606247] [INSPIRE].

[19] A.V. Smirnov, *An Algorithm to construct Grobner bases for solving integration by parts relations*, *JHEP* **04** (2006) 026 [hep-ph/0602078] [INSPIRE].

[20] R.N. Lee, *Group structure of the integration-by-part identities and its application to the reduction of multiloop integrals*, *JHEP* **07** (2008) 031 [arXiv:0804.3008] [INSPIRE].

[21] J. Gluza, K. Kajda and D.A. Kosower, *Towards a Basis for Planar Two-Loop Integrals*, *Phys. Rev. D* **83** (2011) 045012 [arXiv:1009.0472] [INSPIRE].

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

[23] H. Ita, *Two-loop Integrand Decomposition into Master Integrals and Surface Terms*, *Phys. Rev. D* **94** (2016) 116015 [arXiv:1510.05626] [INSPIRE].

[24] K.J. Larsen and Y. Zhang, *Integration-by-parts reductions from unitarity cuts and algebraic geometry*, *Phys. Rev. D* **93** (2016) 041701 [arXiv:1511.01071] [INSPIRE].

[25] J. Böhm, A. Georgoudis, K.J. Larsen, M. Schulze and Y. Zhang, *Complete sets of logarithmic vector fields for integration-by-parts identities of Feynman integrals*, *Phys. Rev. D* **98** (2018) 025023 [arXiv:1712.09737] [INSPIRE].

[26] A. von Manteuffel, E. Panzer and R.M. Schabinger, *Cusp and collinear anomalous dimensions in four-loop QCD from form factors*, *Phys. Rev. Lett.* **124** (2020) 162001 [arXiv:2002.04617] [INSPIRE].

[27] A. von Manteuffel and R.M. Schabinger, *A novel approach to integration by parts reduction*, *Phys. Lett. B* **744** (2015) 101 [arXiv:1406.4513] [INSPIRE].

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

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

[30] J. Klappert, S.Y. Klein and F. Lange, *Interpolation of Dense and Sparse Rational Functions and other Improvements in* ***FireFly***, arXiv:2004.01463 [INSPIRE].

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

[32] J. Böhm, A. Georgoudis, K.J. Larsen, H. Schönemann and Y. Zhang, *Complete integration-by-parts reductions of the non-planar hexagon-box via module intersections*, *JHEP* **09** (2018) 024 [arXiv:1805.01873] [INSPIRE].

[33] D. Bendle et al., *Integration-by-parts reductions of Feynman integrals using Singular and GPI-Space*, *JHEP* **02** (2020) 079 [arXiv:1908.04301] [INSPIRE].

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

[35] H. Frellesvig, F. Gasparotto, M.K. Mandal, P. Mastrolia, L. Mattiazzi and S. Mizera, *Vector Space of Feynman Integrals and Multivariate Intersection Numbers*, *Phys. Rev. Lett.* **123** (2019) 201602 [arXiv:1907.02000] [INSPIRE].

[36] H. Frellesvig et al., *Decomposition of Feynman Integrals on the Maximal Cut by Intersection Numbers*, *JHEP* **05** (2019) 153 [arXiv:1901.11510] [INSPIRE].

[37] H. Frellesvig et al., *Decomposition of Feynman Integrals by Multivariate Intersection Numbers*, arXiv:2008.04823 [INSPIRE].

[38] X. Liu, Y.-Q. Ma and C.-Y. Wang, *A Systematic and Efficient Method to Compute Multi-loop Master Integrals*, *Phys. Lett. B* **779** (2018) 353 [arXiv:1711.09572] [INSPIRE].

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

[40] X. Guan, X. Liu and Y.-Q. Ma, *Complete reduction of integrals in two-loop five-light-parton scattering amplitudes*, *Chin. Phys. C* **44** (2020) 093106 [arXiv:1912.09294] [INSPIRE].

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

[42] Y. Wang, Z. Li and N. Ul Basat, *Direct reduction of multiloop multiscale scattering amplitudes*, *Phys. Rev. D* **101** (2020) 076023 [arXiv:1901.09390] [INSPIRE].

[43] D.A. Kosower, *Direct Solution of Integration-by-Parts Systems*, *Phys. Rev. D* **98** (2018) 025008 [arXiv:1804.00131] [INSPIRE].

[44] A.V. Smirnov and V.A. Smirnov, *How to choose master integrals*, *Nucl. Phys. B* **960** (2020) 115213 [arXiv:2002.08042] [INSPIRE].

[45] J. Usovitsch, *Factorization of denominators in integration-by-parts reductions*, arXiv:2002.08173 [INSPIRE].

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

[47] J.M. Henn, *Lectures on differential equations for Feynman integrals*, *J. Phys. A* **48** (2015) 153001 [arXiv:1412.2296] [INSPIRE].

[48] E.K. Leinartas, *Factorization of rational functions of several variables into partial fractions*, *Izv. Vyssh. Uchebn. Zaved. Mat.* **10** (1978) 47 [*Soviet Math. (Iz. VUZ)* **22** (1978) 35].

[49] A. Raichev, *Leinartas's partial fraction decomposition*, arXiv:1206.4740.

[50] C. Meyer, *Algorithmic transformation of multi-loop master integrals to a canonical basis with CANONICA*, *Comput. Phys. Commun.* **222** (2018) 295 [arXiv:1705.06252] [INSPIRE].

[51] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, B. Page and V. Sotnikov, *Analytic Form of the Planar Two-Loop Five-Parton Scattering Amplitudes in QCD*, *JHEP* **05** (2019) 084 [arXiv:1904.00945] [INSPIRE].

[52] W. Decker, G.-M. Greuel, G. Pfister and H. Schönemann, SINGULAR *4-1-3 — A computer algebra system for polynomial computations*, http://www.singular.uni-kl.de (2020).

[53] A.V. Smirnov and A.V. Petukhov, *The Number of Master Integrals is Finite*, *Lett. Math. Phys.* **97** (2011) 37 [arXiv:1004.4199] [INSPIRE].

[54] O. Gituliar and V. Magerya, *Fuchsia and master integrals for splitting functions from differential equations in QCD*, PoS(LL2016)030 [arXiv:1607.00759] [INSPIRE].

[55] M. Prausa, *epsilon: A tool to find a canonical basis of master integrals*, *Comput. Phys. Commun.* **219** (2017) 361 [arXiv:1701.00725] [INSPIRE].

[56] R.N. Lee, *Reducing differential equations for multiloop master integrals*, *JHEP* **04** (2015) 108 [arXiv:1411.0911] [INSPIRE].

[57] P. Wasser, *Analytic properties of Feynman integrals for scattering amplitudes*, Ph.D. Thesis, Johannes Gutenberg University of Mainz, Mainz Germany (2018).

[58] D. Chicherin, T. Gehrmann, J.M. Henn, P. Wasser, Y. Zhang and S. Zoia, *All Master Integrals for Three-Jet Production at Next-to-Next-to-Leading Order*, *Phys. Rev. Lett.* **123** (2019) 041603 [arXiv:1812.11160] [INSPIRE].

[59] J. Chen, X. Xu and L.L. Yang, *Constructing Canonical Feynman Integrals with Intersection Theory*, arXiv:2008.03045 [INSPIRE].

[60] C. Dlapa, J. Henn and K. Yan, *Deriving canonical differential equations for Feynman integrals from a single uniform weight integral*, *JHEP* **05** (2020) 025 [arXiv:2002.02340] [INSPIRE].

[61] A.B. Goncharov, *Multiple polylogarithms, cyclotomy and modular complexes*, *Math. Res. Lett.* **5** (1998) 497 [arXiv:1105.2076] [INSPIRE].

[62] A.B. Goncharov, M. Spradlin, C. Vergu and A. Volovich, *Classical Polylogarithms for Amplitudes and Wilson Loops*, *Phys. Rev. Lett.* **105** (2010) 151605 [arXiv:1006.5703] [INSPIRE].

[63] A.B. Goncharov, *A simple construction of grassmannian polylogarithms*, arXiv:0908.2238.

[64] C. Duhr, H. Gangl and J.R. Rhodes, *From polygons and symbols to polylogarithmic functions*, *JHEP* **10** (2012) 075 [arXiv:1110.0458] [INSPIRE].

[65] C. Duhr, *Mathematical aspects of scattering amplitudes*, in *Theoretical Advanced Study Institute in Elementary Particle Physics: Journeys Through the Precision Frontier: Amplitudes for Colliders (TASI 2014)*, Boulder U.S.A. (2014), pg. 419 [arXiv:1411.7538] [INSPIRE].

[66] D. Cox, J. Little and D. O'Shea, *Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra*, Springer-Verlag, New York U.S.A. (1992).

[67] R. Ehrenborg and G.-C. Rota, *Apolarity and canonical forms for homogeneous polynomials*, *Eur. J. Combin.* **14** (1993) 157.

[68] G.-M. Greuel and G. Pfister, *A singular introduction to commutative algebra*, second extended edition, Springer, Berlin Germany (2008).

[69] J. Böhm, W. Decker, C. Fieker and G. Pfister, *The use of bad primes in rational reconstruction*, *Math. Comp.* **84** (2015) 3013 [arXiv:1207.1651].

[70] J. Böhm, W. Decker, C. Fieker, S. Laplagne and G. Pfister, *Bad primes in computational algebraic geometry*, in *Lecture Notes in Computer Science. Vol. 9725: Mathematical software — ICMS 2016*, Springer, Berlin Germany (2016), pg. 93.

[71] T. Gehrmann and E. Remiddi, *Two loop master integrals for $\gamma * \to 3$ jets: The Nonplanar topologies*, *Nucl. Phys. B* **601** (2001) 287 [hep-ph/0101124] [INSPIRE].

[72] J. Henn, B. Mistlberger, V.A. Smirnov and P. Wasser, *Constructing d-log integrands and computing master integrals for three-loop four-particle scattering*, *JHEP* **04** (2020) 167 [arXiv:2002.09492] [INSPIRE].

[73] S. Abreu, L.J. Dixon, E. Herrmann, B. Page and M. Zeng, *The two-loop five-point amplitude in $\mathcal{N} = 4$ super-Yang-Mills theory*, *Phys. Rev. Lett.* **122** (2019) 121603 [arXiv:1812.08941] [INSPIRE].

[74] D. Chicherin, J. Henn and V. Mitev, *Bootstrapping pentagon functions*, *JHEP* **05** (2018) 164 [arXiv:1712.09610] [INSPIRE].

[75] J. Boehm, W. Decker, A. Frühbis-Krüger, F.-J. Pfreundt, M. Rahn and L. Ristau, *Towards massively parallel computations in algebraic geometry*, *Found. Comput. Math.* (2020) [arXiv:1808.09727].