

P3H-20-008, TTP20-008

## How to choose master integrals

A.V. Smirnov<sup>a,b</sup>, V.A. Smirnov<sup>c,\*</sup>

<sup>a</sup>*Research Computing Center, Moscow State University,  
119992 Moscow, Russia*

<sup>b</sup>*Institut für Theoretische Teilchenphysik, Karlsruhe Institute of Technology (KIT), 76128  
Karlsruhe, Germany*

<sup>c</sup>*Skobeltsyn Institute of Nuclear Physics of Moscow State University,  
119992 Moscow, Russia*

---

### Abstract

The standard procedure when evaluating integrals of a given family of Feynman integrals, corresponding to some Feynman graph, is to construct an algorithm which provides the possibility to write any particular integral as a linear combination of so-called master integrals. To do this, public (AIR, FIRE, REDUZE, LiteRed, KIRA) and private codes based on solving integration by parts relations are used. However, the choice of the master integrals provided by these codes is not always optimal. We present an algorithm to improve a given basis of the master integrals, as well as its computer implementation; see also a competitive variant [1].

*Keywords:* Multiloop Feynman integrals, dimensional regularization, integration by parts relations, master integrals, *arXiv: 2002.08042*

---



---

\*Corresponding author

*Email addresses:* [asmirnov80@gmail.com](mailto:asmirnov80@gmail.com) (A.V. Smirnov), [smirnov@theory.sinp.msu.ru](mailto:smirnov@theory.sinp.msu.ru) (V.A. Smirnov)

## 1. Introduction

After integration by parts (IBP) reduction was invented [2] it became possible to decompose the problem of evaluating Feynman integrals into two parts: a reduction to so-called master integrals (MIs) and the evaluation of these MIs. In the eighties and nineties, the first part of this procedure was solved ‘by hand’ but then computer codes which perform an IBP reduction appeared. At the moment, there are at least five public codes (**AIR**, **FIRE**, **REDUZE**, **LiteRed**, **KIRA**) [3–10], and a number of private codes<sup>1</sup>. By definition, MIs are integrals that appear on the right-hand sides of solutions of IBP relations, so that they form a basis in the linear space of integrals of a given family associated with an  $h$ -loop graph,

$$G_{i_1, \dots, i_L} = \int \dots \int \prod_{l=1}^L \frac{1}{(m_l^2 - p_l^2)^{i_l}} d^d k_1 \dots d^d k_h. \quad (1)$$

Here  $d = 4 - 2\epsilon$  is the dimensional regularization parameter,  $k_1, \dots, k_h$  are loop momenta, and momenta of the lines  $p_l$  are expressed in terms of linear combinations of the loop momenta  $k_i$  and external momenta  $q_j$ . Integrals of a given family are, in particular, functions of indices  $a_i$  (powers of the propagators) which can be considered as integer variables.

The title of the paper might look strange because a set of the MIs is produced automatically after a code to solve IBP relations is applied, so that there is no choice at this point. However, experience tells us that, especially in sufficiently complicated situations, the basis provided by such a code, can be bad because the denominators of the coefficients of MIs in IBP-reductions of input integrals can be quite cumbersome. Of course, coefficients in the decomposition of a given input integral over MIs are always rational functions of everything, i.e. of  $d$  and kinematical invariants, because solving IBP relations reduces to solving sparse linear systems of equations with the help of a variant of the Gaussian elimination. With big denominators, the reduction to the MIs can be rather complicated and, in some cases, even unfeasible, i.e. requiring too much time or/and requiring too much operative memory.

On the other hand, zeros of the denominators in IBP-reductions are connected with singularities of Feynman integrals described by Landau equations, implying that the denominators can be decomposed into a product of functions of kinematical invariants and masses (independent of  $d$ ) and linear functions of  $d$  (independent of other variables). This follows from an analysis of convergence properties of Feynman integrals represented as parametric integrals over Feynman parameters. This analysis can be performed, in some situations, with classical sector decompositions by Hepp and Speer used to prove theorems on

---

<sup>1</sup>Let us observe that, for concrete families of Feynman integrals, specially constructed IBP-reduction programs can be much more powerful than the above mentioned general programs. Here remarkable examples are two public codes **Mincer** [11, 12] and **Forcer** [13] successfully applied for the reduction of three- and four-loop massless propagator diagrams, respectively.

renormalization [14, 15], or, in more general situations, with modern recursive sector decompositions [16–18]. We can thus hope that, with a better choice of the MIs, the denominators on the right-hand side of the IBP reduction relations will be *good*, i.e. decomposed as a product of polynomials of kinematical invariants and masses, independent of  $d$ , and linear terms of the form  $ad + b$  with rational real numbers  $a$  and  $b$ . However, sometimes a factor independent of  $d$  can be a rather cumbersome polynomial of kinematic invariants and masses so that it can be qualified naturally as a bad denominator. We will take into account this possibility in our code described in Section 4.

Let us also call a denominator *bad* if it is not good. We will also call a basis *good* if the denominators in IBP reductions into that basis are good.

We will now explain how one can improve a given basis of the MIs if it is not good<sup>2</sup>. In the next section, we describe an algorithm to improve a given basis of MIs. In Section 3, we discuss possible origins of bad denominators. In Section 4, we present a code based on our algorithm and, in Section 5, we discuss some other ways of improving a given basis of MIs. In the Appendix, we present an example which demonstrates how our code works.

## 2. The algorithm

Suppose we have a basis  $f_i(x, d), i = 1, \dots, N$ , of MIs obtained with some IBP reduction code. For simplicity of presentation, we describe the case of two scales, where  $x$  is their ratio, for example,  $x = q^2/m^2$ . Let us check whether it is a good or bad basis and if it is bad let us try to improve it. Let us run an IBP reduction code on a set of sample integrals taken from all the sectors with non-zero numbers of the MIs. In our calculations, we prefer to choose integrals with indices 0, 1 and 2: we include in the sample list corner integrals of these sectors, i.e. without indices equal to two, then integrals with one index equal to two, then integrals with two indices index equal to two. (In complicated situations, sample integrals with three indices equal to two or even higher might be also needed.) We prefer sample integrals without negative indices because, according to our experience, the choice of MIs with negative indices has more chances to lead to an appearance of bad denominators. Moreover, symmetries of Feynman integrals are more visible for integrals without negative indices. However, the following algorithm and its implementation work for any set of sample integrals.

Anyway, we start with an IBP reduction of a set of the sample integrals and know their reduction which can be written in the form of a list of substitutions:

$$f(x, d) \rightarrow \sum_{i=1}^N c_i(x, d) f_i(x, d) , \quad (2)$$

where the coefficients  $c_i$  are rational functions of  $x$  and  $d$ .

---

<sup>2</sup>In our experience, we already improved bases of MIs in many calculations without developing a code for this and we believe that other people also did this. At least we know one example in the literature [19].

Let us call by the *level* the number of positive indices of an integral. Let us analyze reductions of the sample integrals starting from sectors of the minimal level. Suppose that we are at the lowest level where bad denominators appear. We now determine which sectors are responsible for the generation of these bad denominators by analyzing at which of the MIs of the given level the bad denominators appear. We now consider these sectors one by one.

For a given sector, let  $g_j, j = 1, 2, \dots$ , be the set of the corresponding sample integrals. Their reduction has the form

$$g_j = \sum_{i=1}^{s_i} c_{j,i} f_i + \dots, \quad (3)$$

where  $f_i, i = 1, 2, \dots, s_i$  are MIs of the given sector and dots stand for the contribution of lower sectors. For a given  $g_j$ , analyze numerators of those coefficients  $c_{j,i}$  which involve the current bad denominator. Let us consider a numerator bad or good using the same definition as formulated above for the denominators.

(a) *A simple situation.* Suppose that for some  $i$ , the numerator is good. Then replace the MI  $f_i$  by the new MI  $g_j$ . After using an explicit relation between  $f_i$  and  $g_j$  which is found by solving a linear equation, find the mapping which expresses  $f_i$  in terms of  $g_i$  and the other MIs. Check that after this change, the current bad denominator disappears. When  $f_i$  is written down in terms of  $g_i$  and other MIs, the bad denominator goes to the numerator and cancels bad denominators also in other places, while the numerator in  $c_{j,i}$  goes to the denominator but it is harmless because it is good.

(b) *A more complicated situation.* Suppose now that for all  $i$ , the corresponding numerators are bad. Choose  $i$  such that the length (defined as the number of terms in the expanded expression) of the numerator is minimal and make the corresponding replacement. Therefore, the resulting bad denominators become better, but they are not yet good. Repeat this procedure until all denominators are good.

Now perform this procedure also for other sectors of the given level, then proceed to higher levels eventually reaching the top sector. As a result we obtain a list of desirable MIs. Within FIRE, this list is encoded via the option **preferred** in subsequent reductions.

To speed up the analysis of the bad denominators, one can fix either  $d$  or other variables. In particular, in situations with many kinematic invariants, one can get rid of non-linear denominators in  $d$  fixing all the other parameters and thereby make the sample reduction much faster.

### 3. Where do the bad denominators appear from?

In order to use the code efficiently and not to expect it to do things that it is not designed to do, it is important to understand how the bad denominators appear. To our understanding, bad denominators can appear because either

1. the current choice of MIs is not the optimal choice, or,

2. the current set of MIs is not minimal so that there is a hidden relation between them.

It is important to understand which of those variants (or both) is the case to improve properly the current basis of MIs.

But what actually is this variant 2? If MIs are irreducible how can one have a relation between them? The answer lies in the implementation of reduction programs. There can be a relation between "MIs" produced by a reduction program that the reduction program cannot reveal, and there are reasons for this, one of which is that it might be that not all relations between Feynman integrals follow from IBP relations. In fact, it is an open question whether they follow or not. However modern reduction programs normally try to use symmetries in addition to the IBP relations. For example, FIRE can use internal sector symmetries from LiteRed (depending on the `#pos_pref` option), and so we normally do not miss relations for MIs in a single sector.

Still, there is a reason why extra relations can be missed due to the way in which reduction programs are implemented. (The statement is valid for FIRE, but we expect it is also to be valid for other reduction programs.) The programs work sector by sector, so if during reduction a relation is reduced completely out of a sector, relating only integrals of lower sectors, reduction programs tend to drop relations of this sort at this point. Therefore we locate a possible source of relations between MIs of lower sectors that reduction programs can ignore.

This leads to the following conclusion: the variants can be distinguished one from the other. In case there is a relation  $r$  between MIs of level  $l$ , it means that there should be a relation (IBP or symmetry) of a level higher than  $l$ , that could be reduced and lead to  $r$ . It also means that the analysis of bad denominators at level  $l$  won't reveal such a relation. The bad denominators of type 2 are revealed only when one takes a reduction relation for an integral of a level higher than  $l$ , and the bad coefficients are those at integrals of level  $l$ .

On the other hand, the bad denominators of type 1 appear inside a level, when considering coefficients of integrals of the same level on the left-hand side and right-hand side of the formula. Still due to the way we order Feynman integrals (trying to reduce to lower sectors), there will be bad denominators at lower levels as well.

It is important to note, that the code described here only aims at a good basis choice and is able to get rid of bad denominators of the first type. In case there are extra relations between MIs, one needs another way to decrease their number, and this will be discussed in Section 5, however even in this case the code can improve the basis.

However this consideration has another important consequence that might be useful for the application of the code: while searching for bad denominators of type 1, one can consider only coefficients expressing integrals of a given level by MIs of the same level. Everything that is below can be dropped for the purpose of finding a good basis.

#### 4. The code

The above algorithm is implemented in Mathematica as a part of FIRE, starting from the public release 6.4.1 (with more options in 6.4.2), however all functions related to this algorithm are placed in a separate Mathematica file `mm/ImproveMasters.m`, and moreover it can be used not only together with FIRE, but also with other reduction programs.

Let us explain the format used by the algorithm. First, there is a Feynman integral, which is defined by a problem number  $pn$  (a positive integer) and a set of indices. Like in other parts of FIRE, we use the following form for a Feynman integral:

$$G[pn, \{i_1, i_2, \dots, i_n\}] \quad (4)$$

Then let us define a ‘relation’, i.e. a representation of one integral as a linear combination of other integrals. Of course this could be simply a Mathematica rule with a linear combination on the right-hand side, but for optimization reasons we prefer to store a relation in a structured format, where the right-hand side of the rule (Mathematica *Rule*) comes as a list of pairs containing an integral and a coefficient each:

$$G[pn, \{i_1, i_2, \dots, i_n\}] \rightarrow \{\{G[pn, \{j_{1,1}, \dots, j_{1,n}\}], c_1\}, \dots, \{G[pn, \{j_{m,1}, \dots, j_{m,n}\}], c_m\}\} \quad (5)$$

The problem number should be always the same, the integrals on the right-hand side should not be repeated. This format is much more convenient for algorithmic reasons because one does not need to separate coefficients from the right-hand sides all the time. It is also not difficult to convert between the traditional format with a sum and the structured format. We provide a function *RelationSum2List* that converts a rule with a sum to a rule with a list. The inverse conversion is even less complex and can be obtained with *Rule[##[[1]], Plus@@Times@@@##[[2]]]&*, but we also for convenience we provide the *RelationList2Sum* function.

The input for the main algorithm is a Mathematica list of relations. This format can be obtained in FIRE with the

$$Tables2Rules[filename, Identity, False] \quad (6)$$

command. Here *Identity* stands for no function application to coefficients, this will be done by the algorithm later anyway. The last parameter *False* stands for *JoinTerms = False* meaning that we are not going to convert the expressions from the list format to the sum format.

The main function provided by the algorithm is

$$ImproveMasters[relations, level] \quad (7)$$

or, starting from version 6.4.2,

$$\text{ImproveMasters}[\text{relations}, \text{level}, \text{length}] \quad (8)$$

Here *level* stands for the level of integrals (number of positive indices) in which the code will work, and *length* is the minimal length of a polynomial independent of *d* starting from which it is considered bad. For example, *length* = 10 can be a reasonable choice.

The intermediate output of the code is self-explanatory; it prints the bad denominator factors found, the sectors in which they are found, lists of MIs involved and the replacements of MIs it makes in order to get rid of bad denominators. The output is a pair containing a new set of relations and the good basis of MIs. The list of MIs in the output contains only MIs in sectors where a change was required. This set can be used as the set of preferred MIs in FIRE in subsequent IBP reductions or in a similar way in other reduction programs.

To find bad factors FIRE uses *Together* to simplify the fractions, then uses the *Denominator* function to get the denominators and then calls the provided *FindBadFactorsInCoefficient* function that first uses *FactorList* and then analyses the factors. The condition for the factor to be "bad" is a set either depending both on *d* and other variables or containing a non-linear dependence on *d*.

As explained in the previous section, the search for a good basis can be performed purely inside a given level, dropping everything that is below. Therefore if one is not immediately interested in new relations but is searching for the list of preferred integrals only, it is safe to restrict the search to a given level, and that can improve performance of the code greatly. To do that one can use the function *LevelPart*[*relations*, *level*] that keeps only the current level part. For example, one can call

$$\text{ImproveMasters}[\text{LevelPart}[\text{relations}, \text{level}], \text{level}][[2]] \quad (9)$$

to get the list of preferred integrals for the current level. Note that the *LevelPart* function not only picks the relations for integrals of a given level (which could be done, for example, with

*Select*[*relations*, (*IntegralLevel*[*First*[*##*] == *level*) &]) but also leaves only integrals of the desired level on the right-hand side.

The code comes with a number of auxiliary functions. The

$$\text{BadRelationParts}[\text{expr}_-, \text{level}_- : 0, \text{onlyCurrent}_- : \text{False}] \quad (10)$$

function picks out only parts of the original rules where coefficients contain a bad factor in the denominator. If *level* is non-zero, then it considers only integrals of the specified level on the left-hand sides, and if also *onlyCurrent* is set to *True*, then it also filters the right-hand sides to have only integrals of the same level. This might be useful since only coefficients at the current level are related to the search of a good basis. In case the function *ImproveBasis* succeeds, the result of *BadRelationParts*[*expr*, *level*, *True*] on the first part of its return value be an empty set.

If one is only interested in displaying bad factors, ignoring the integrals with which they appear, one can use the

$$\text{FindBadFactorsInRules}[\text{expr}_-, \text{level}_- : 0, \text{onlyCurrent}_- : \text{False}] \quad (11)$$

function with the same parameters. Similarly, if the function *ImproveBasis* succeeds, the result of *FindBadFactorsInRules*[*expr*, *level*, *True*] on the first part of its return value should be an empty set.

The purpose of the code is in searching for a good basis, and this is related only to coefficients of a given level. As explained above there can be extra relations between MIs that usually lead to bad coefficients of MIs of a lower level than the integral on the left-hand side. Since the code *ImproveBasis* cannot help with getting rid of them, alternative methods should be used that are discussed briefly below. However if one has such an extra relation, it can also be applied with the structured rules format. This is done with the *SubstituteRuleIntoRules*[*rules*\_, *rule*\_] function.

In case there are no more bad denominators remaining, both functions *BadRelationParts*[*expr*] and *FindBadFactorsInRules*[*expr*] return empty sets.

## 5. Discussion and conclusion

We have explained how an IBP reduction can be simplified due to a better choice of the MIs. As mentioned above, another source of bad denominators can be hidden relations between a current set of the MIs. When presenting the release of FIRE4 [20] we suggested a way to find such relations. It is based on symmetries of Feynman integrals of a given family. One finds relations between sample integrals (we prefer to consider integrals with indices equal to 0, 1 and 2, with the number of indices = 2 equal to one, two, three and, in some complicated cases, even higher, however the code can work with any sample choice). Within FIRE, this can be done with the help of the command **FindRules** but there are also other ways to find such symmetry relations. Then one performs an IBP reduction of these sample integrals and checks whether symmetry relations simply yield identities or produce new relations between current MIs.

In fact, one can check whether the number of the MIs in a given sector is minimal using the code **Mint** [21] based on algebraic geometry. If the code gives a number which is less than the number of current MIs in the given sector then it is quite reasonable to look for a hidden relation. However, additional relations obtained with the help of symmetries usually provide relations in partially overlapping sectors while **Mint** provides information about a fixed sector.

We know also examples where running an IBP reduction with KIRA and FIRE and equating the corresponding results provides a missing relation.

Getting rid of bad denominators is important not only because it improves reduction performance with respect to runtimes and memory usage. In particular, when applying approaches based on modular arithmetic (**Finred** [22] and **FIRE** [5]), it is necessary, first, to reveal the form of possible denominators.



Moreover, within the method of differential equations, it is important to get rid of denominators which are spurious and can be eliminated by a basis change.

Let us emphasize that our algorithm can be applied not only with FIRE but also with other reduction programs. On the other hand, one more tool for improving a given basis of MIs is described in a ‘parallel’ paper [1] by an author of KIRA.

## Appendix A. An example

Let us see how bad denominators can be eliminated by our code for the family of integrals associated with the three-loop vertex graph shown in Fig. A.1. In

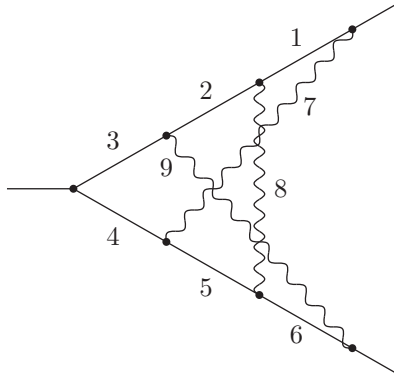


Figure A.1: A three-loop vertex graph. Solid lines are with the mass  $m$ , wavy lines are massless.

accordance with Eq. (1), the squares of momenta  $p_l$  in the propagators are

$$\begin{aligned} &\{-(q_1 + k_1)^2 + m^2, -(q_1 + k_1 + k_2)^2 + m^2, -(q_1 + k_1 + k_2 + k_3)^2 + m^2, \\ &\quad -(q_2 + k_1 + k_2 + k_3)^2 + m^2, -(q_2 + k_2 + k_3)^2 + m^2, -(q_2 + k_3)^2 + m^2, \\ &\quad -k_1^2, -k_2^2, -k_3^2, -(k_1 - k_2)^2, -(k_1 - k_3)^2, -(k_2 - k_3)^2\} , \end{aligned}$$

where  $k_i$  are loop momenta and  $q_i$  are external momenta with  $q_1^2 = m^2, q_2^2 = m^2, (q_1 - q_2)^2 = s$ . The first nine indices can be positive while the last three indices are always non-positive and stand only for numerators.

As a list of sample integrals we choose the list described in Section 2 without negative indices and the number of indices equal to 2 up to two. Looking for hidden relations between primary MIs, according to the procedure mentioned in the beginning of Section 5 and based on symmetries, we find the following relations between MIs in partially overlapping sectors:

$$G_{0,1,0,0,1,2,1,0,0,0,0,0} \rightarrow \frac{d-2}{8m^2} G_{0,0,0,1,0,1,1,1,0,0,0,0} - \frac{2d-5}{4m^2} G_{0,1,0,0,1,1,1,0,0,0,0,0} .$$

Tables obtained with FIRE for sample integrals can be downloaded from <http://theory.sinp.msu.ru/~smirnov/imi>. There is also the set of the MIs

which we obtain after improving a primary basis. Running the code described in Section 4 at levels 3 and 4 shows no bad denominators. At level 5, there are several bad denominators. To get rid of the bad denominator

$$448m^4 - 240dm^4 + 32d^2m^4 - 580m^2s + 320dm^2s - 44d^2m^2s + 268s^2 - 150ds^2 + 21d^2s^2$$

we have to make a change in the sector with the following MIs:

$$\{G_{0,1,0,1,1,0,0,1,1,0,0,0}, G_{0,1,0,1,1,0,0,1,2,0,0,0}, G_{0,1,0,1,1,0,0,2,1,0,0,0}, G_{0,1,0,1,1,0,0,2,2,0,0,0}, G_{0,1,0,1,2,0,0,1,1,0,0,0}, G_{0,1,0,1,2,0,0,1,2,0,0,0}, G_{0,1,0,2,1,0,0,1,1,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,0,1,2,0,0,1,1,0,0,0}$  instead of  $G_{0,1,0,1,1,0,0,2,2,0,0,0}$ .

To get rid of the bad denominator  $12m^2 - 4dm^2 - 16s + 5ds$  we have to make a change in the sector with the following MIs:

$$\{G_{0,0,1,0,1,0,1,1,1,0,0,0}, G_{0,0,1,0,1,0,1,1,2,0,0,0}, G_{0,0,1,0,1,0,2,1,1,0,0,0}, G_{0,0,1,0,2,0,1,1,1,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,0,2,0,1,0,1,1,1,0,0,0}$  instead of  $G_{0,0,1,0,1,0,1,1,1,0,0,0}$ .

To get rid of the bad denominator  $16m^2 - 4dm^2 - 10s + 3ds$  we have to make a change in the sector with

$$\{G_{0,1,0,1,1,1,0,0,1,0,0,0}, G_{0,1,0,1,1,1,0,0,2,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,0,1,1,1,0,0,1,0,0,0}$  instead of  $G_{0,1,0,1,1,1,0,0,2,0,0,0}$ .

To get rid of the bad denominator  $28m^2 - 8dm^2 - 4s + ds$  we have to make a change in the sector with

$$\{G_{0,1,1,1,1,0,0,1,0,0,0,0}, G_{0,1,1,1,1,0,0,2,0,0,0,0}, G_{0,1,1,1,2,0,0,1,0,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,2,1,1,0,0,1,0,0,0,0}$  instead of  $G_{0,1,1,1,1,0,0,1,0,0,0,0}$ .

To get rid of the bad denominator  $4m^2 + 4s - ds$  we have to make a change in the sector with

$$\{G_{0,1,0,0,1,1,1,0,2,0,0,0}, G_{0,1,0,0,1,1,2,0,1,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,0,0,1,1,1,0,1,0,0,0}$  instead of  $G_{0,1,0,0,1,1,1,0,2,0,0,0}$ .

At level 6, there is a very bad denominator

$$\begin{aligned} &1166901120m^6 - 2228472576dm^6 + 1889043552d^2m^6 - 934622944d^3m^6 \\ &+ 298051104d^4m^6 - 63674944d^5m^6 + 9133728d^6m^6 - 850048d^7m^6 \\ &+ 46656d^8m^6 - 1152d^9m^6 - 1077693120m^4s + 2033788512dm^4s \\ &- 1698824192d^2m^4s + 825710264d^3m^4s - 257891900d^4m^4s + 53810944d^5m^4s \\ &- 7523356d^6m^4s + 681848d^7m^4s - 36472d^8m^4s + 880d^9m^4s \\ &+ 323477760m^2s^2 - 602689792dm^2s^2 + 493601656d^2m^2s^2 - 233171456d^3m^2s^2 \\ &+ 69995306d^4m^2s^2 - 13842696d^5m^2s^2 + 1802914d^6m^2s^2 - 149054d^7m^2s^2 \end{aligned}$$

$$\begin{aligned}
& +7094d^8m^2s^2 - 148d^9m^2s^2 - 31819200s^3 + 58544600ds^3 \\
& -46779452d^2s^3 + 21211150d^3s^3 - 5972893d^4s^3 + 1070399d^5s^3 - \\
& 119334d^6s^3 + 7576d^7s^3 - 210d^8s^3
\end{aligned}$$

which is generated by the sector with ten MIs

$$\begin{aligned}
& \{G_{0,1,1,0,1,1,1,1,0,0,0,0}, G_{0,1,1,0,1,1,1,2,0,0,0,0}, G_{0,1,1,0,1,1,2,1,0,0,0,0}, G_{0,1,1,0,1,1,2,2,0,0,0,0}, \\
& G_{0,1,1,0,1,2,1,1,0,0,0,0}, G_{0,1,1,0,1,2,1,2,0,0,0,0}, G_{0,1,1,0,1,2,2,1,0,0,0,0}, G_{0,1,1,0,2,1,1,1,0,0,0,0}, \\
& G_{0,1,2,0,1,1,1,1,0,0,0,0}, G_{0,2,1,0,1,1,1,1,0,0,0,0}\}.
\end{aligned}$$

The code does not find a replacement that immediately removes this denominator. Then the code looks for variants of reducing the length of this bad denominator. The best variant corresponds to a reduction of the length from 35 to 16 and is achieved by choosing  $G_{0,1,2,0,2,1,1,1,0,0,0,0}$  instead of  $G_{0,1,1,0,1,2,1,2,0,0,0,0}$ .

Then the code takes care of the bad denominator

$$\begin{aligned}
& 52992m^4 - 47136dm^4 + 16784d^2m^4 - 3136d^3m^4 + 328d^4m^4 - 16d^5m^4 \\
& -30480m^2s + 24512dm^2s - 7308d^2m^2s + 1036d^3m^2s - 82d^4m^2s \\
& +4d^5m^2s + 4200s^2 - 3140ds^2 + 774d^2s^2 - 63d^3s^2
\end{aligned}$$

in the previous sector where one of the MIs was replaced. This denominator is eliminated by choosing  $G_{0,2,1,0,1,1,2,1,0,0,0,0}$  instead of  $G_{0,1,1,0,1,1,2,2,0,0,0,0}$ .

To get rid of the bad denominator  $3480m^4 - 1860dm^4 + 244d^2m^4 - 2872m^2s + 1528dm^2s - 200d^2m^2s + 390s^2 - 207ds^2 + 27d^2s^2$  we have to make a change in the sector with

$$\begin{aligned}
& \{G_{0,1,0,1,1,0,1,1,1,0,0,0}, G_{0,1,0,1,1,0,1,1,2,0,0,0}, G_{0,1,0,1,1,0,1,2,1,0,0,0}, G_{0,1,0,1,1,0,2,1,1,0,0,0}, \\
& G_{0,1,0,1,2,0,1,1,1,0,0,0}, G_{0,1,0,2,1,0,1,1,1,0,0,0}, G_{0,2,0,1,1,0,1,1,1,0,0,0}\}.
\end{aligned}$$

This can be achieved by choosing  $G_{0,2,0,1,2,0,1,1,1,0,0,0}$  instead of  $G_{0,1,0,1,1,0,1,1,2,0,0,0}$ .

To get rid of the bad denominator  $16m^2 - 4dm^2 - 10s + 3ds$  we have to make a change in the two sectors with

$$\begin{aligned}
& \{G_{1,0,0,1,1,1,0,1,1,0,0,0}, G_{1,0,0,1,1,1,0,1,2,0,0,0}, G_{1,0,0,1,1,1,0,2,2,0,0,0}, G_{1,0,0,1,2,1,0,1,1,0,0,0}, \\
& G_{1,1,0,0,1,1,1,0,1,0,0,0}, G_{1,1,0,0,1,1,1,0,2,0,0,0}, G_{1,1,0,0,1,1,2,0,2,0,0,0}, G_{1,1,0,0,1,2,1,0,1,0,0,0}\}.
\end{aligned}$$

This can be achieved by choosing  $G_{2,0,0,2,1,1,0,1,1,0,0,0}$  instead of  $G_{1,0,0,1,1,1,0,2,2,0,0,0}$  and  $G_{2,2,0,0,1,1,1,0,1,0,0,0}$  instead of  $G_{1,1,0,0,1,1,2,0,2,0,0,0}$ .

To get rid of the bad denominator  $22m^2 - 6dm^2 - 15s + 3ds$  we have to make changes in the two sectors with

$$\{G_{0,1,0,0,1,1,1,1,1,0,0,0}, G_{0,1,0,0,1,1,1,2,1,0,0,0}, G_{1,0,0,1,0,1,1,1,1,0,0,0}, G_{1,0,0,1,0,1,2,1,1,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,1,0,0,2,1,1,1,1,0,0,0}$  instead of  $G_{0,1,0,0,1,1,1,1,1,0,0,0}$  and  $G_{2,0,0,1,0,1,1,1,1,0,0,0}$  instead of  $G_{1,0,0,1,0,1,1,1,1,0,0,0}$ .

To get rid of the bad denominator  $28m^2 - 18s + 3ds$  we have to make changes in the two sectors with

$$\{G_{0,1,1,1,1,0,0,1,1,0,0,0}, G_{0,1,1,1,1,0,0,1,2,0,0,0}, G_{1,0,1,1,0,1,0,1,1,0,0,0}, G_{1,0,1,1,0,1,0,1,2,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,1,1,1,0,0,1,1,0,0,0}$  instead of  $G_{0,1,1,1,1,0,0,1,2,0,0,0}$  and  $G_{1,0,1,1,0,2,0,1,1,0,0,0}$  instead of  $G_{1,0,1,1,0,1,0,1,2,0,0,0}$ .

To get rid of the bad denominator  $8m^2 - 2dm^2 - 7s + 2ds$  we have to make a change in the sector with

$$\{G_{0,1,0,1,0,1,1,1,1,0,0,0}, G_{0,1,0,1,0,1,1,1,2,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,0,1,0,1,1,1,1,0,0,0}$  instead of  $G_{0,1,0,1,0,1,1,1,2,0,0,0}$ .

At level 7, there is a bad denominator  $4480m^6 - 4192dm^6 + 1280d^2m^6 - 128d^3m^6 - 1040m^4s + 1208dm^4s - 456d^2m^4s + 56d^3m^4s + 120m^2s^2 - 152dm^2s^2 + 66d^2m^2s^2 - 10d^3m^2s^2 + 4ds^3 - 4d^2s^3 + d^3s^3$ . We make a change in the sector with

$$\{G_{0,1,1,0,1,1,1,1,1,0,0,0}, G_{0,1,1,0,1,1,1,1,2,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,1,1,0,2,1,1,1,1,0,0,0}$  instead of  $G_{0,1,1,0,1,1,1,1,2,0,0,0}$ .

To get rid of the bad denominator  $360m^4 - 152dm^4 + 16d^2m^4 + 16s^2 - 8ds^2 + d^2s^2$  we have to make a change in the sector with

$$\{G_{1,0,1,1,0,1,1,1,1,0,0,0}, G_{1,0,1,1,0,1,1,1,2,0,0,0}, G_{1,0,1,1,0,1,1,2,1,0,0,0}, \\ G_{1,0,1,1,0,1,1,2,2,0,0,0}, G_{1,0,1,1,0,2,1,1,1,0,0,0}\}.$$

This can be achieved by choosing  $G_{2,0,1,1,0,1,2,1,1,0,0,0}$  instead of  $G_{1,0,1,1,0,1,1,2,2,0,0,0}$ .

To get rid of the bad denominator  $360m^4 - 152dm^4 + 16d^2m^4 + 16s^2 - 8ds^2 + d^2s^2$  we have to make a change in the sector with

$$\{G_{1,0,1,1,0,1,1,1,1,0,0,0}, G_{1,0,1,1,0,1,1,1,2,0,0,0}, G_{1,0,1,1,0,1,1,2,1,0,0,0}, \\ G_{1,0,1,1,0,1,1,2,2,0,0,0}, G_{1,0,1,1,0,2,1,1,1,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,1,1,0,1,1,1,1,0,0,0}$  instead of  $G_{0,1,1,1,0,1,1,1,1,0,0,0}$ .

At level 8, one bad denominator appears,  $8m^4 - 4dm^4 + 24m^2s - 5dm^2s - 5s^2 + ds^2$ . We have to make changes in the two sectors with the following MIs:

$$\{G_{0,1,1,1,1,1,1,1,1,0,0,0}, G_{0,1,1,1,1,1,1,1,2,0,0,0}, G_{0,1,1,1,1,1,1,1,2,1,0,0,0}, G_{0,1,1,1,1,2,1,1,1,0,0,0}, \\ G_{1,0,1,1,1,1,1,1,1,0,0,0}, G_{1,0,1,1,1,1,1,1,2,0,0,0}, G_{1,0,1,1,1,1,2,1,1,0,0,0}, G_{1,0,1,1,2,1,1,1,1,0,0,0}\}.$$

This can be achieved by choosing  $G_{0,2,1,1,1,1,1,1,1,0,0,0}$  instead of  $G_{0,1,1,1,1,1,1,1,1,0,0,0}$  and  $G_{2,0,1,1,1,1,1,1,1,0,0,0}$  instead of  $G_{1,0,1,1,1,1,1,1,1,0,0,0}$ .

Finally, our code reveals no bad denominators at level 9.

*Acknowledgments.* We are grateful to J. Davies, M. Steinhauser and J. Uschitsch for fruitful discussions and various pieces of advice. The work is carried out according to the research program of Moscow Center of Fundamental and Applied Mathematics.

## References

- [1] J. Usovitsch, Factorization of denominators in integration-by-parts reductions, [arXiv:2002.08173](#).
- [2] K. G. Chetyrkin, F. V. Tkachov, Integration by Parts: The Algorithm to Calculate beta Functions in 4 Loops, *Nucl. Phys. B* 192 (1981) 159–204. doi:10.1016/0550-3213(81)90199-1.
- [3] C. Anastasiou, A. Lazopoulos, Automatic integral reduction for higher order perturbative calculations, *JHEP* 07 (2004) 046. [arXiv:hep-ph/0404258](#), doi:10.1088/1126-6708/2004/07/046.
- [4] A. V. Smirnov, FIRE5: a C++ implementation of Feynman Integral REduction, *Comput. Phys. Commun.* 189 (2015) 182–191. [arXiv:1408.2372](#), doi:10.1016/j.cpc.2014.11.024.
- [5] A. V. Smirnov, F. S. Chuharev, FIRE6: Feynman Integral REduction with Modular Arithmetic, *Comput. Phys. Commun.* 247. [arXiv:1901.07808](#), doi:10.1016/j.cpc.2019.106877.
- [6] A. von Manteuffel, C. Studerus, Reduze 2 - Distributed Feynman Integral Reduction, [arXiv:1201.4330](#).
- [7] R. N. Lee, Presenting LiteRed: a tool for the Loop InTEgrals REDuction, [arXiv:1212.2685](#).
- [8] R. N. Lee, LiteRed 1.4: a powerful tool for reduction of multiloop integrals, *J.Phys.Conf.Ser.* 523 (2014) 012059. [arXiv:1310.1145](#), doi:10.1088/1742-6596/523/1/012059.
- [9] Maierhöfer, Philipp and Usovitsch, Johann and Uwer, Peter, Kira — A Feynman integral reduction program, *Comput. Phys. Commun.* 230 (2018) 99–112. [arXiv:1705.05610](#), doi:10.1016/j.cpc.2018.04.012.
- [10] Maierhöfer, Philipp and Usovitsch, Johann, Kira 1.2 Release Notes, [arXiv:1812.01491](#).
- [11] S. A. Larin, F. V. Tkachov, J. A. M. Vermaseren, The FORM version of MINCER.
- [12] S. G. Gorishnii, S. A. Larin, L. R. Surguladze, F. V. Tkachov, MINCER: Program for multiloop calculations in quantum field theory for the SCHOONSCHIP system, *Comput. Phys. Commun.* 55 (1989) 381–408. doi:10.1016/0010-4655(89)90134-3.
- [13] B. Ruijl, T. Ueda, J. A. M. Vermaseren, Forcer, a FORM program for the parametric reduction of four-loop massless propagator diagrams, [arXiv:1704.06650](#).

- [14] K. Hepp, Proof of the Bogolyubov-Parasiuk theorem on renormalization, Commun. Math. Phys. 2 (1966) 301–326. doi:10.1007/BF01773358.
- [15] E. R. Speer, Ultraviolet and Infrared Singularity Structure of Generic Feynman Amplitudes, Ann. Inst. H. Poincaré Phys. Theor. 23 (1975) 1–21.
- [16] T. Binoth, G. Heinrich, An automatized algorithm to compute infrared divergent multi-loop integrals, Nucl. Phys. B585 (2000) 741–759. arXiv:hep-ph/0004013, doi:10.1016/S0550-3213(00)00429-6.
- [17] C. Bogner, S. Weinzierl, Resolution of singularities for multi-loop integrals, Comput. Phys. Commun. 178 (2008) 596–610. arXiv:0709.4092, doi:10.1016/j.cpc.2007.11.012.
- [18] A. V. Smirnov, M. N. Tentyukov, Feynman Integral Evaluation by a Sector decomposition Approach (FIESTA), Comput. Phys. Commun. 180 (2009) 735–746. arXiv:0807.4129, doi:10.1016/j.cpc.2008.11.006.
- [19] K. Melnikov, L. Tancredi, C. Wever, Two-loop  $gg \rightarrow Hg$  amplitude mediated by a nearly massless quark, JHEP 11 (2016) 104. arXiv:1610.03747, doi:10.1007/JHEP11(2016)104.
- [20] A. V. Smirnov, V. A. Smirnov, FIRE4, LiteRed and accompanying tools to solve integration by parts relations, Comput. Phys. Commun. 184 (2013) 2820–2827. arXiv:1302.5885, doi:10.1016/j.cpc.2013.06.016.
- [21] R. N. Lee, A. A. Pomeransky, Critical points and number of master integrals, JHEP 11 (2013) 165. arXiv:1308.6676, doi:10.1007/JHEP11(2013)165.
- [22] A. von Manteuffel, R. M. Schabinger, A novel approach to integration by parts reduction, Phys. Lett. B744 (2015) 101–104. arXiv:1406.4513, doi:10.1016/j.physletb.2015.03.029.