

Towards the next Kira release

Fabian Lange,^{a,b,*} Johann Usovitsch^c and Zihao Wu^d

^a*Physik-Institut, Universität Zürich,
Winterthurerstrasse 190, 8057 Zürich, Switzerland*

^b*Paul Scherrer Institut,
5232 Villigen PSI, Switzerland*

^c*Theoretical Physics Department, CERN,
1211 Geneva, Switzerland*

^d*School of Fundamental Physics and Mathematical Science, Hangzhou Institute of Advanced Study, UCAS,
Hangzhou 310024, China*

*E-mail: fabian.lange@physik.uzh.ch, johann.usovitsch@cern.ch,
wuzihao@mail.ustc.edu.cn*

The reduction of Feynman integrals to a basis of master integrals plays a crucial role for many high-precision calculations and Kira is one of the leading tools for this task. In these proceedings we discuss some of the new features and improvements currently being developed for the next release.

*Loops and Legs in Quantum Field Theory (LL2024)
14-19, April, 2024
Wittenberg, Germany*

*Speaker

1. Introduction

Feynman integrals play a central role in precision calculations, both in the Standard Model and beyond. A direct integration of all integrals of an amplitude, nowadays often many tens or hundreds of thousands, is usually unfeasible. The default strategy in modern precision calculations is thus to first reduce the integrals to a much smaller set of master integrals employing integration-by-parts (IBP) identities [1, 2] and the Laporta algorithm [3]. This strategy is implemented in the public programs AIR [4], FIRE [5–9], Reduze [10, 11], Kira [12, 13], FINITEFLOW [14] in combination with LiteRed [15, 16], and most recently Blade [17].

After an overview of IBP reductions and Kira in Section 2, we discuss some of the features and improvements that we are developing for the next release in Section 3. We highlight the performance improvements with some benchmarks in Section 4 and discuss our findings and future directions in Section 5.

2. An overview over integration-by-parts reductions and Kira

In general, a scalar L -loop Feynman integral can be written as

$$T_j \equiv T(a_1, \dots, a_N) = \int \left(\prod_{i=1}^L d^d \ell_i \right) \frac{1}{P_1^{a_1} P_2^{a_2} \dots P_N^{a_N}}, \quad (1)$$

where

$$P_j = q_j^2 - m_j^2, \quad j = 1, \dots, N \quad (2)$$

are inverse Feynman propagators. The momenta q_j are linear combinations of the loop momenta ℓ_i , $i = 1, \dots, L$, and external momenta p_k , $k = 1, \dots, E$ for $E + 1$ external legs (or $E = 0$ for vacuum integrals), and m_j are the propagator masses. The a_j are the (integer) propagator powers. The set of inverse propagators must be complete and independent in the sense that every scalar product of momenta can be uniquely expressed as a linear combination of the P_j , squared masses m_j^2 , and external kinematical invariants. The number of propagators is thus $N = \frac{L}{2}(L + 2E + 1)$ including auxiliary propagators that only appear with $a_j \leq 0$.

The Feynman integrals of Eq. (1) are in general not independent. In Refs. [1, 2] it was found that they are related by so-called integration-by-parts (IBP) identities

$$\int \left(\prod_{i=1}^L d^d \ell_i \right) \frac{\partial}{\partial \ell_i^\mu} \frac{q_j^\mu}{P_1^{a_1} P_2^{a_2} \dots P_N^{a_N}} = 0, \quad (3)$$

where q_j^μ is either a loop momentum, an external momentum, or a linear combination thereof. In addition to these IBP identities, there also exist so-called Lorentz-invariance identities [18]

$$\sum_{i=1}^E \left(p_i^\nu \frac{\partial}{\partial p_{i\mu}} - p_i^\mu \frac{\partial}{\partial p_{i\nu}} \right) T(a_1, \dots, a_N) = 0. \quad (4)$$

They do not provide additional information, but it was found that they often facilitate the reduction process. Finally, there are symmetry relations between Feynman integrals which are most often

searched for by comparing graph polynomials with Pak's algorithm [19]. All these relations provide linear equations of the form

$$0 = \sum_i c_i(\{s_j, m_k\})T(a_{1,i}, \dots, a_{N,i}) \quad (5)$$

between different Feynman integrals, where the coefficients $c_i(\{s_j, m_k\})$ are rational functions of the kinematic invariants s_j and the propagator masses m_k . The relations can be solved to express the Feynman integrals in terms of a basis of master integrals. It was even shown that the number of master integrals for the standard Feynman integrals of Eq. (1) is finite [20].

Nowadays, the most prominent solution strategy is the Laporta algorithm [3]: The linear relations of Eq. (5) are generated for different values of the propagator powers a_j resulting in a system of equations between specific Feynman integrals. Each choice of $\{a_j\}$ is called a *seed* and this generation process is also known as *seeding*. The system can then be solved with Gauss-type elimination algorithms.

This requires an ordering of the integrals. In Kira we first assign the integrals to so-called topologies based on their respective sets of propagators, i.e. their momenta and masses. We assign a unique integer number to each topology. Secondly, we assign each integral to a sector

$$S = \sum_{j=1}^N 2^{j-1} \theta(a_j - \frac{1}{2}), \quad (6)$$

where $\theta(x)$ is the Heaviside step function. A sector S is called subsector of another sector S' (with propagators powers a'_j) if $S < S'$ and $a_j \leq a'_j$ for all $j = 1, \dots, N$. We denote as top-level sectors those sectors which are not subsectors of other sectors that contain Feynman integrals occurring in the reduction problem at hand. For the discussions in the following sections it is beneficial to use the big-endian binary notation instead of the sector number defined by Eq. (6), i.e. the sector $S = 5$ is represented by b1010 where each number represents one bit of the number 5. In this notation it is immediately visible that b1010 is a subsector of b1110.

Furthermore, it is useful to define the number of propagators with positive powers,

$$t = \sum_{j=1}^N \theta(a_j - \frac{1}{2}), \quad (7)$$

and

$$r = \sum_{j=1}^N a_j \theta(a_j - \frac{1}{2}), \quad s = - \sum_{j=1}^N a_j \theta(\frac{1}{2} - a_j), \quad d = \sum_{j=1}^N (a_j - 1) \theta(a_j - \frac{1}{2}) \quad (8)$$

denoting the sum of all positive powers, the negative sum of all negative powers, and the sum of positive powers larger than 1, respectively. These concepts allow us to order the integrals in Kira. Secondly, they are used to limit the seeds $\{a_j\}$ for which equations are generated. We generate the equations only for those sets $\{a_j\}$ for which $r \leq r_{\max}$, $s \leq s_{\max}$, and $d \leq d_{\max}$, where r_{\max} , s_{\max} , and d_{\max} are input provided by the user.

The workflow in Kira is split into two components: first generating the system of equations and then solving it. We offer two different strategies for the latter. First, the system can be solved

analytically using the computer algebra system Fermat [21] for the rational function arithmetic. Secondly, the system can be solved using modular arithmetic over finite fields [22–25]. The variables, i.e. kinematic invariants and masses, are replaced by random integer numbers and all arithmetic operations are performed over prime fields. Choosing the largest 63-bit primes allows us to perform all operations on native data types and to avoid large intermediate expressions. The final result can then be interpolated and reconstructed by *probing* the system sufficiently many times for different random choices of the variables. We employ the library FireFly for this task [26, 27].

The main improvements in the upcoming version of Kira discussed in Section 3 concern the generation of the system of equations. Let us thus outline the current algorithm in more detail:

1. Identify trivial sectors and discard all integrals in these sectors.
2. Identify symmetries between sectors including those of other integral families. Drop sectors which can identically be mapped away. The remaining symmetries give rise to additional equations to be generated.
3. For every remaining sector starting from the simplest one, generate all equations within the specifications provided by the user:
 - (a) Generate one equation.
 - (b) Check if it is linearly independent by inserting all previously generated equations and solving them numerically using modular arithmetic [23].
 - (c) Keep it if it is linearly independent, otherwise drop it.
4. After all equations were generated, solve the system for the target integrals specified by the user with modular arithmetic and drop all equations which do not contribute to their solutions.

At the end of this procedure Kira has generated a system of linearly independent equations specifically trimmed to suffice to reduce the target integrals to master integrals.

3. Improvements and new features under development

In this section we discuss some of the improvements and new features which are currently under development.

3.1 Internal reordering of propagators

It is well known that the definition of the propagators in Eq. (2) influences the performance of the reduction. Not only different choices for the momenta have an impact, simply reordering them can make a difference, especially when selecting equations before the actual reduction as in Kira. This can be demonstrated with the four-loop propagator type topology depicted in Fig. 1. We reduce all integrals appearing in the quark self energy amplitude for the relation between the quark mass renormalization constants in the $\overline{\text{MS}}$ and the on-shell schemes [28]. We compare propagator order 1,

$$p_1^2 - m_1^2, (p_1 - q)^2, \underline{(p_2 + p_3)^2}, (p_1 - p_3 + p_4 - q)^2, (p_3 - p_4)^2, \underline{(p_3 + q)^2}, \underline{(p_2 + p_4)^2}, \quad (9)$$

$$p_2^2 - m_2^2, (-p_1 + p_2 + p_3 - p_4 + q)^2 - m_2^2, p_3^2, p_4^2, \underline{(p_1 + p_3)^2}, \underline{(p_1 + p_4)^2}, \underline{(p_1 + p_2)^2},$$

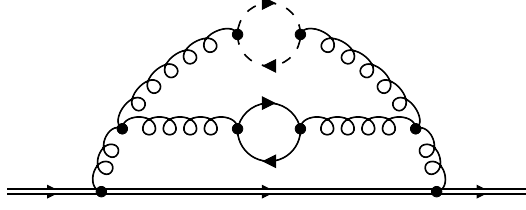


Figure 1: Four-loop propagator topology with two masses.

which was automatically chosen when generating the amplitude in Ref. [28], with propagator order 2,

$$\begin{aligned}
 & p_3^2, p_4^2, p_1^2 - m_1^2, p_2^2 - m_2^2, (p_1 - q)^2, (p_3 - p_4)^2, (p_1 - p_3 + p_4 - q)^2, \\
 & (-p_1 + p_2 + p_3 - p_4 + q)^2 - m_2^2, \underline{(p_2 + p_3)^2}, \underline{(p_3 + q)^2}, \underline{(p_2 + p_4)^2}, \underline{(p_1 + p_3)^2}, \\
 & \underline{(p_1 + p_4)^2}, \underline{(p_1 + p_2)^2},
 \end{aligned} \quad (10)$$

which follows the guidelines outlined in Ref. [29]: order by the number of momenta and put massless propagators first. The underlined entries denote auxiliary propagators which only appear in the numerator and should be placed at the end. When comparing the systems of equations generated with Kira 2.3 for the two orders in Table 1, we see that the system generated with order 2 contains 10 % less equations, 14 % less terms, and can be solved 28 % faster than the system generated with order 1.

Table 1: Comparison of the generated and selected systems of equations for the two propagator orders defined in Eqs. (9) and (10) of the four-loop propagator topology depicted in Fig. 1.

	order 1	order 2
# of equations	3 737 521	3 362 446
# of terms	47 122 072	40 326 035
Probe time	57 s	41 s

Since version 2.3 Kira offers the possibility to reorder the propagators internally independent of the order specified in the integral family definition, i.e. the propagator order in the input and output files differs from the order Kira uses internally to perform the reduction. The ordering can either be chosen manually or be adjusted automatically according to four different ordering schemes. The one outlined above usually provides the best results. With the next release of Kira we will automatically turn on this ordering scheme by default.

3.2 Improved seeding

The choice of which equations should be generated for the reduction has severe impact on the performance of the reduction. Kira seeds conservatively and generates all equations within the bounds specified by the user to reduce to the minimal basis of master integrals. As outlined in Section 2, irrelevant equations are filtered out through a selection procedure. However, for integrals

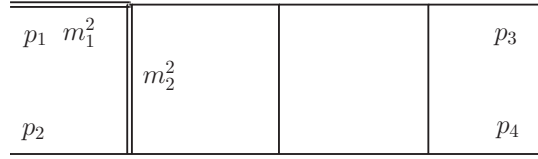


Figure 2: The planar double box `topo7` contributing, e.g., to single-top production.

with many loops or high values of r , s , and d the combinatorics of the seeds easily gets out of hand and already makes the generation of the system of equations unfeasible, running both into runtime and memory limits. We thus revised the seeding process in `Kira` and identified three areas for improvement: seeding of sectors related by symmetries, seeding of sectors containing preferred master integrals, and seeding of subsectors. We address the three areas in the following.

In the `Kira 2.3`, sectors which are related by symmetries to other sectors are seeded with the same values for r_{\max} , s_{\max} , and d_{\max} . Similarly, sectors which contain one of the preferred master integrals are seeded with the maximum values of r_{\max} , s_{\max} , and d_{\max} from all sectors. Both choices were made to ensure to reduce to the minimal basis, but turn out to be too conservative and negatively impacting performance. In the upcoming release those sectors will be seeded with the default values for subsectors.

Currently the values for r_{\max} , s_{\max} , and d_{\max} set for a sector are propagated down to all subsectors. For example, if we are interested in the integral $T(1, 1, 1, 1, 1, 1, 1, 0, -4)$, the typical choice is to seed the sector `b111111100` with $r_{\max} = 7$ and $s_{\max} = 4$. This then propagates to the lower sectors and sector `b111111000` is automatically seeded with $r_{\max} = 7$, $s_{\max} = 4$, and $d_{\max} = 1$ and sector `b111000000` with $r_{\max} = 7$, $s_{\max} = 4$, and $d_{\max} = 4$, respectively. The increase in d_{\max} can already be restricted by manually setting d_{\max} , but we previously argued against this to prevent finding too many master integrals. With the new release we revise this recommendation and encourage setting d_{\max} . However, keeping s_{\max} constant when descending to the subsectors is even more problematic because the combinatorics of distributing irreducible scalar products in lower sectors grows quickly. As it turns out, most of these seeds are actually not required to reduce integrals with positive s in higher sectors and one can decrease s_{\max} in lower sectors. With the upcoming release we provide an option to restrict s in lower sectors by $s \leq (t - l + 1)$, where t was defined in Eq. (7) and l can be chosen by the user. This was first described in Ref. [30]. The same observation was also made by the developers of `Blade` [17] and `FIRE` [31] and is already available in the former code.

3.3 Improved integral selection

The algorithm to select relevant equations in `Kira 2.3` is not optimal. One weakness can easily be illustrated when reducing all integrals in the top-level sector `b111111100` of the topology `topo7` depicted in Fig. 2 with exactly $s = 5$. First, we seed with $s_{\max} = 5$ and, secondly, with $s_{\max} = 6$, but in both cases select the equations for the same set of integrals. As can be seen in Table 2, the number of selected equations and with it the number of terms as well as the time to solve the system increases. This can be explained by the fact that `Kira` generates and selects sector by sector. Thus, it prefers high seeds of low sectors over low seeds of higher sectors. However,

Table 2: Comparison of the selected systems for `topo7` when seeded with different values of s_{\max} . The improved seeding discussed in Section 3.2 has been disabled as much as possible without major code modifications.

	2.3, $s_{\max} = 5$	2.3, $s_{\max} = 6$	dev, $s_{\max} = 5$	dev, $s_{\max} = 6$
# of selected equations	62 514	85 119	34 908	34 908
# of terms	461 336	659 803	245 975	245 975
Probe time	0.16 s	0.24 s	0.084 s	0.084 s

higher seeds most often result in more complicated equations.

Secondly, we identified another problem: Kira selects equations which do not contribute to the final result because they cancel in intermediate steps. This can be illustrated with the toy system

$$\begin{aligned}
 \text{(i)} \quad & 0 = x[6] + b_1 x[2], \\
 \text{(ii)} \quad & 0 = x[6] + c_1 x[1], \\
 \text{(iii)} \quad & 0 = x[5] + a_1 x[2] + a_2 x[4] + a_3 x[3], \\
 \text{(iv)} \quad & 0 = x[5] + a_1 x[2],
 \end{aligned} \tag{11}$$

where $x[i]$ represents a Feynman integral and the Roman number in front denotes the equation number. We now solve this system using Gaussian elimination. After the forward elimination the system reads

$$\begin{aligned}
 \text{(i)} \quad & 0 = b_1 x[2] - c_1 x[1] \quad \{\text{ii}\}, \\
 \text{(ii)} \quad & 0 = x[6] + c_1 x[1] \quad \{\}, \\
 \text{(iii)} \quad & 0 = a_2 x[4] + a_3 x[3] \quad \{\text{iv}\}, \\
 \text{(iv)} \quad & 0 = x[5] + a_1 x[2] \quad \{\},
 \end{aligned} \tag{12}$$

where the numbers in the curly brackets denote which equations have been inserted. After the back substitution we obtain

$$\begin{aligned}
 \text{(i)} \quad & 0 = b_1 x[2] - c_1 x[1] \quad \{\text{ii}\}, \\
 \text{(ii)} \quad & 0 = x[6] + c_1 x[1] \quad \{\}, \\
 \text{(iii)} \quad & 0 = a_2 x[4] + a_3 x[3] \quad \{\text{iv}\}, \\
 \text{(iv)} \quad & 0 = x[5] + a_1 c_1 / b_1 x[1] \quad \{\text{i, ii}\}.
 \end{aligned} \tag{13}$$

Kira 2.3 selects equations after the back substitution using the information in the curly brackets of Eq. (13). If we are interested in the integral $x[4]$, it thus selects equation (iii) because it provides the solution for $x[4]$, then equation (iv) which was inserted into equation (iii), and finally equations (i) and (ii) which were inserted into equation (iv). However, the latter two do not provide any information: it is easy to spot in Eq. (11) that equation (iii) contains equation (iv) as a subequation. This subequation thus vanishes after inserting equation (iv) only and the information of equations (i) and (ii) is not needed. We call this phenomenon *hidden zero* which in general can take more complicated forms.

So far we were not able to develop a systematic algorithm to detect all of the hidden zeros. For the upcoming release we decided to instead rely on heuristics: We select the equations for the target

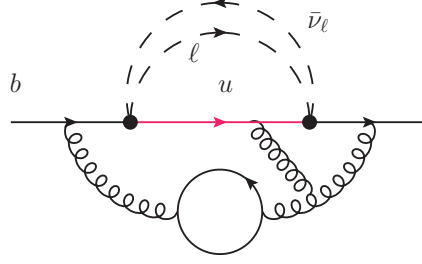


Figure 3: Diagram contributing to the α_s^3 corrections of the semileptonic $b \rightarrow u$ decay.

integrals after the forward elimination only and check if they contain all information to reduce to the basis of master integrals. All non-reduced integrals are then added to the target integrals and we repeat the process until all target integrals are reduced to master integrals.

Returning to Table 2, we see that this improved selection algorithm addresses both problems described in this section: The development version of Kira selects the same equations even if seeding with higher values of s_{\max} and less equations overall, indicating that we got rid of some hidden zeros. Note that the improved seeding discussed in Section 3.2 has been disabled as much as possible without major code modifications, but a thorough study of the improved selection alone is left for the future.

3.4 Symbolic propagator powers

With the upcoming version of Kira it will be possible to perform reductions with symbolic propagator powers by marking the associated propagators. One application are the α_s^3 corrections to the semileptonic $b \rightarrow u$ decay considered in Ref. [32]. In the sample diagram shown in Fig. 3 the lepton-neutrino loop does not receive QCD corrections and can simply be integrated out using

$$\int d^d p \frac{p^{\mu_1} \dots p^{\mu_N}}{(-p^2)[-(p-q)^2]} = \frac{i\pi^{2-\epsilon}}{(-q^2)^\epsilon} \times \sum_{i=0}^{[N/2]} f(\epsilon, i, N) \left(\frac{q^2}{2}\right)^i \{[g]^i [q]^{N-2i}\}^{\mu_1 \dots \mu_N}, \quad (14)$$

where the propagator $-q^2$ is now raised to the power ϵ . Therefore, this trick reduces the five-loop reduction to a four-loop one at the price of one symbolic propagator power which can be handled by the upcoming version of Kira.

4. Benchmarks

In this section we show two benchmarks to compare the current development version of Kira to the most recent release version 2.3.

4.1 Can one tune Kira 2.3 already?

Our first example is the vertex topology shown in Fig. 4 appearing in the heavy-to-light quark form factors at the three-loop level [33]. We want to reduce all integrals of the amplitude in R_ξ gauge. The curious reader might ask if it is already possible to achieve some of the improvements discussed in Section 3 by carefully tuning the input parameters in Kira 2.3, because they are just

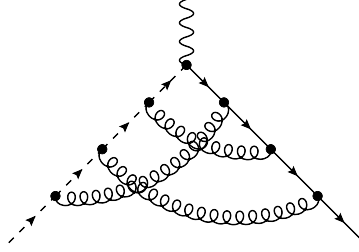


Figure 4: Topology contributing to the heavy-to-light quark form factors at the three-loop level.

Table 3: Comparison of the systems of equations for the topology shown in Fig. 4 with the recommended settings of Kira 2.3, an attempt to tune Kira 2.3, and the development version.

	Kira 2.3	Kira 2.3, tuned	Kira dev
# of generated equations	304 821 183	198 405 903	3 779 262
# of selected equations	21 666 745	24 708 949	828 010
# of terms	828 060 148	583 976 288	12 516 161
Time to generate and select	96 359 s	59 579 s	1 328 s
Memory to generate and select	194 GiB	129 GiB	3.2 GiB
Probe time	2 255 s	3 726 s	19 s



Figure 5: Massless double pentagon topology.

updated recommendations, e.g. setting d_{\max} . We tried this to the best of our abilities, and failed as evident in Table 3. While it is already possible to generate less equations compared to the previously recommended strategy, unfortunately this increases the number of selected equations and the time to solve the system significantly. On the other hand, with the current development version we achieve a success on all fronts: Compared to Kira 2.3 with recommended settings, we generate 81 times less equations and select 26 times less equations which are overall simpler and contain 66 times less terms. The generation and selection speed up by a factor 73 and require 61 times less memory. Finally, the time to solve the system numerically over a finite field reduces by a factor of 119.

4.2 Double pentagon

As the second benchmark we choose the massless double pentagon topology shown in Fig. 5. We reduce all integrals in the top-level sector b1111111000 with $s = 5$. Again we observe impressive improvements shown in Table 4: We generate 222 times less equations, select 28 times

Table 4: Comparison of the systems of equations for the double pentagon topology shown in Fig. 5 with the recommended settings of Kira 2.3 and the development version. In the last column we reduce the latter system with RATRACER.

	Kira 2.3	Kira dev	Kira dev + RATRACER
# of generated equations	16 872 564	76 045	-
# of selected equations	1 157 381	41 998	-
# of terms	23 053 485	734 833	-
Time to generate and select	1 259 s	14 s	-
Memory to generate and select	30 GiB	1.9 GiB	-
Probe time	38 s	1.2 s	0.12 s

less equations, and the system contains 31 times less terms. This speeds up the generation and selection by a factor 90 and requires 16 times less memory. The probe time decreases by a factor of 32.

Additionally, we use the program RATRACER [34] to reduce the system generated by the development version. It records the trace of operations performed on the input values and, thus, gets rid of the overhead associated to solving the system. This way the probe time is sped up by another factor of 10 which puts it well below 1 s, fast enough to sample the phase space as for example done in Ref. [35].

5. Discussion

In these proceedings we presented some of the improvements and new features that we are developing for the next release of the Feynman integral reduction program Kira. Besides the support for symbolic propagator powers, we focused on improvements to the algorithms to seed and select equations. Our benchmarks show a significant speed-up and relaxation of the memory requirements by one or two orders of magnitude for the selected projects. We already successfully applied the new version of Kira to three vastly different projects [30, 32, 33].

However, further investigation is in order: So far, we only quantified the numbers for the generation of the system and the reduction using modular arithmetics over finite fields, neglecting the CPU time for the interpolation and reconstruction algorithms. While this is negligible for small rational functions with only a few variables, it might dominate the runtime for complicated rational functions. In this case there will be lesser improvements to the performance. Thus, both technical and algorithmic improvements in FireFly [26, 27] might be necessary, e.g. following the idea of Ref. [36]. On the other hand it might be worthwhile to sample the phase space for $2 \rightarrow 3$ processes instead of interpolating the reduction tables, as for example pursued in Ref. [35]. As shown Section 4.2, the performance of the new version of Kira looks promising for this strategy, especially in combination with RATRACER [34] or potentially the strategy presented in Ref. [37].

Furthermore, we did not investigate yet how much the improvements enhance the algebraic reduction with Fermat [21]. In this context it will also be interesting to study how other computer algebra systems might improve the performance [9, 38].

Finally, we are looking into extending the symmetry finder algorithms following the ideas presented in Ref. [39] which will allow us to map more sectors away and reduce the number of master integrals.

Acknowledgments

The work of F.L. was supported by the Swiss National Science Foundation (SNSF) under contract [TMSG12_211209](#). Some of the Feynman diagrams were drawn with the help of Axodraw [40] and JaxoDraw [41] and some with the help of FeynGame [42, 43].

References

- [1] F.V. Tkachov, *A theorem on analytical calculability of 4-loop renormalization group functions*, *Phys. Lett. B* **100** (1981) 65.
- [2] 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.
- [3] S. Laporta, *High-precision calculation of multiloop Feynman integrals by difference equations*, *Int. J. Mod. Phys. A* **15** (2000) 5087 [[hep-ph/0102033](#)].
- [4] C. Anastasiou and A. Lazopoulos, *Automatic integral reduction for higher order perturbative calculations*, *JHEP* **07** (2004) 046 [[hep-ph/0404258](#)].
- [5] A.V. Smirnov, *Algorithm FIRE – Feynman Integral REduction*, *JHEP* **10** (2008) 107 [[0807.3243](#)].
- [6] A.V. Smirnov and V.A. Smirnov, *FIRE4, LiteRed and accompanying tools to solve integration by parts relations*, *Comput. Phys. Commun.* **184** (2013) 2820 [[1302.5885](#)].
- [7] A.V. Smirnov, *FIRE5: A C++ implementation of Feynman Integral REduction*, *Comput. Phys. Commun.* **189** (2015) 182 [[1408.2372](#)].
- [8] A.V. Smirnov and F.S. Chukharev, *FIRE6: Feynman Integral REduction with modular arithmetic*, *Comput. Phys. Commun.* **247** (2020) 106877 [[1901.07808](#)].
- [9] A.V. Smirnov and M. Zeng, *FIRE 6.5: Feynman integral reduction with new simplification library*, *Comput. Phys. Commun.* **302** (2024) 109261 [[2311.02370](#)].
- [10] C. Studerus, *Reduze – Feynman integral reduction in C++*, *Comput. Phys. Commun.* **181** (2010) 1293 [[0912.2546](#)].
- [11] A. von Manteuffel and C. Studerus, *Reduze 2 - Distributed Feynman Integral Reduction*, [1201.4330](#).
- [12] P. Maierhöfer, J. Usovitsch and P. Uwer, *Kira—A Feynman integral reduction program*, *Comput. Phys. Commun.* **230** (2018) 99 [[1705.05610](#)].

- [13] J. Klappert, F. Lange, P. Maierhöfer and J. Usovitsch, *Integral reduction with Kira 2.0 and finite field methods*, *Comput. Phys. Commun.* **266** (2021) 108024 [2008.06494].
- [14] T. Peraro, *FINITEFLOW: multivariate functional reconstruction using finite fields and dataflow graphs*, *JHEP* **07** (2019) 031 [1905.08019].
- [15] R.N. Lee, *Presenting LiteRed: a tool for the Loop InTEgrals REDuction*, 1212.2685.
- [16] R.N. Lee, *LiteRed 1.4: a powerful tool for reduction of multiloop integrals*, *J. Phys. Conf. Ser.* **523** (2014) 012059 [1310.1145].
- [17] X. Guan, X. Liu, Y.-Q. Ma and W.-H. Wu, *Blade: A package for block-triangular form improved Feynman integrals decomposition*, 2405.14621.
- [18] T. Gehrmann and E. Remiddi, *Differential equations for two-loop four-point functions*, *Nucl. Phys. B* **580** (2000) 485 [hep-ph/9912329].
- [19] A. Pak, *The toolbox of modern multi-loop calculations: novel analytic and semi-analytic techniques*, *J. Phys. Conf. Ser.* **368** (2012) 012049 [1111.0868].
- [20] A.V. Smirnov and A.V. Petukhov, *The Number of Master Integrals is Finite*, *Lett. Math. Phys.* **97** (2011) 37 [1004.4199].
- [21] R.H. Lewis, *Computer Algebra System Fermat*, <https://home.bway.net/lewis>.
- [22] M. Kauers, *Fast Solvers for Dense Linear Systems*, *Nucl. Phys. B Proc. Suppl.* **183** (2008) 245.
- [23] P. Kant, *Finding linear dependencies in integration-by-parts equations: A Monte Carlo approach*, *Comput. Phys. Commun.* **185** (2014) 1473 [1309.7287].
- [24] A. von Manteuffel and R.M. Schabinger, *A novel approach to integration by parts reduction*, *Phys. Lett. B* **744** (2015) 101 [1406.4513].
- [25] T. Peraro, *Scattering amplitudes over finite fields and multivariate functional reconstruction*, *JHEP* **12** (2016) 030 [1608.01902].
- [26] J. Klappert and F. Lange, *Reconstructing rational functions with FireFly*, *Comput. Phys. Commun.* **247** (2020) 106951 [1904.00009].
- [27] J. Klappert, S.Y. Klein and F. Lange, *Interpolation of dense and sparse rational functions and other improvements in FireFly*, *Comput. Phys. Commun.* **264** (2021) 107968 [2004.01463].
- [28] M. Fael, F. Lange, K. Schönwald and M. Steinhauser, *A semi-analytic method to compute Feynman integrals applied to four-loop corrections to the $\overline{\text{MS}}$ -pole quark mass relation*, *JHEP* **09** (2021) 152 [2106.05296].
- [29] P. Maierhöfer and J. Usovitsch, *Kira 1.2 Release Notes*, 1812.01491.

- [30] M. Driesse, G.U. Jakobsen, G. Mogull, J. Plefka, B. Sauer and J. Usovitsch, *Conservative Black Hole Scattering at Fifth Post-Minkowskian and First Self-Force Order*, *Phys. Rev. Lett.* **132** (2024) 241402 [2403.07781].
- [31] Z. Bern, E. Herrmann, R. Roiban, M.S. Ruf, A.V. Smirnov, V.A. Smirnov et al., *Amplitudes, Supersymmetric Black Hole Scattering at $O(G^5)$, and Loop Integration*, [2406.01554](#).
- [32] M. Fael and J. Usovitsch, *Third order correction to semileptonic $b \rightarrow u$ decay: Fermionic contributions*, *Phys. Rev. D* **108** (2023) 114026 [2310.03685].
- [33] M. Fael, T. Huber, F. Lange, J. Müller, K. Schönwald and M. Steinhauser, *Heavy-to-light form factors to three loops*, [2406.08182](#).
- [34] V. Magerya, *Rational Tracer: a Tool for Faster Rational Function Reconstruction*, [2211.03572](#).
- [35] B. Agarwal, G. Heinrich, S.P. Jones, M. Kerner, S.Y. Klein, J. Lang et al., *Two-loop amplitudes for $t\bar{t}H$ production: the quark-initiated N_f -part*, *JHEP* **05** (2024) 013 [2402.03301].
- [36] H.A. Chawdhry, *p -adic reconstruction of rational functions in multi-loop amplitudes*, [2312.03672](#).
- [37] X. Liu, *Reconstruction of rational functions made simple*, *Phys. Lett. B* **850** (2024) 138491 [2306.12262].
- [38] K. Mokrov, A. Smirnov and M. Zeng, *Rational Function Simplification for Integration-by-Parts Reduction and Beyond*, [2304.13418](#).
- [39] Z. Wu and Y. Zhang, *A new method for finding more symmetry relations of Feynman integrals*, [2406.20016](#).
- [40] J.A.M. Vermaseren, *Axodraw*, *Comput. Phys. Commun.* **83** (1994) 45.
- [41] D. Binosi and L. Theußl, *JaxoDraw: A Graphical user interface for drawing Feynman diagrams*, *Comput. Phys. Commun.* **161** (2004) 76 [hep-ph/0309015].
- [42] R.V. Harlander, S.Y. Klein and M. Lipp, *FeynGame*, *Comput. Phys. Commun.* **256** (2020) 107465 [2003.00896].
- [43] R. Harlander, S.Y. Klein and M.C. Schaaf, *FeynGame-2.1 – Feynman diagrams made easy*, *PoS EPS-HEP2023* (2024) 657 [2401.12778].