PROCEEDINGS OF SCIENCE

New multiloop capabilities of FeynCalc 10

Vladyslav Shtabovenko[a,*]

[a] *Vladyslav Shtabovenko*
Center for Particle Physics Siegen (CPPS), Theoretische Physik 1,
Universität Siegen, Walter-Flex-Str. 3,
57068 Siegen, Germany

E-mail: shtabovenko@physik.uni-siegen.de

We briefly introduce new multiloop capabilities of the MATHEMATICA package FEYNCALC 10 and a collection of interfaces connecting FEYNCALC to such popular tools as QGRAF, FIESTA, PYSECDEC, LOOPTOOLS, KIRA, FIRE or FERMAT. In addition to that, we showcase the application of these codes to the ongoing study of the "soft-overlap" contribution to $B_c \to \eta_c$ transition form factors at large hadronic recoil.

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

*Speaker

PoS(LL2024)071

1. Introduction

Multiloop calculations belong to the fields of research that heavily depend on software tools and the available computational resources. Obviously, the majority of steps necessary to calculate a Feynman diagram involves repetitive algebraic manipulations that are highly amenable to automation. However, the number of the required manipulations as well as the number of Feynman diagrams at higher perturbative orders tend to grow so fast, that any attempt to approach such calculations in a naive way will inevitably break down. This is why the development of public codes implementing state-of-the-art algorithms and streamlining the common steps required to perform higher-order perturbative calculations is crucial to enable further progress in the field.

While automation of tree- and one-loop-level calculations has been successfully pushed forward since several decades (cf. e. g. refs. [1–18]), the multi-loop case still remains more challenging. Despite some progress in the last years [19–26], even at two loops a fully generic code for automatic cross-section and decay rate calculations is still out of reach.

On the one hand, we have plethora of publicly available tools for handling different steps of multiloop Feynman diagram evaluation including a powerful symbolic manipulation system FORM [27, 28] capable of handling expressions containing millions of terms. On the other hand, full automation of such calculations in a manner similar to e. g. MadGraph or FormCalc is still far from being attainable due to many technical challenges accompanying such endeavors.

Partial automation, i. e. generation of Feynman diagrams, their algebraic simplification and the reduction of loop integrals to masters, is something that has been done by the practitioners since many years. Unfortunately, in most cases no usable codes were made public. Some famous programs such as q2e and exp from Karlsruhe [29, 30] are not freely downloadable but at least available upon request.

In recent years this situation started to change with many frameworks addressing automatic calculations of multiloop amplitudes being made available to the wider public under open source licenses. Some notable examples are Alibrary[1], tapir [31], FeAmGen.jl [32], HepLib [33, 34] or MaRTIn [35]. This is undoubtedly a very positive development in our field that has potential to make such calculations more accessible to the vast majority of particle theorists.

In this proceeding we would like to report on another tool that falls into this category, known under the name of FeynCalc [36–40]. Unlike most other multiloop codes that were written from scratch, this Mathematica package has been known to the community for almost 35 years. Initially developed as a tool for one-loop calculations, it gradually evolved towards higher loops, culminating in the official release of FeynCalc 10 at the end 2023. In addition to that, we have developed a collection of interfaces connecting FeynCalc to useful multiloop-related programs such as QGRAF [41], FIRE [42–44], KIRA [45–49], FIESTA [50, 51], pySecDec [52] or FERMAT [53]. This FeynHelpers add-on for FeynCalc has not yet been officially released but is already publicly available[2] and properly documented.

This report is organized as follows. In Section 2 we describe the implementation of FeynCalc's new multiloop capabilities and briefly mention the related routines. The FeynHelpers add-on is covered in Section 3, while Section 4 showcases a practical application of this technology in the

[1] https://magv.github.io/alibrary/
[2] https://github.com/FeynCalc/feynhelpers

context of Soft-Collinear-Effective-Theory (SCET) [54–57]. Finally, in Section 5 we summarize the current state of affairs.

2. FeynCalc

The most crucial step in making FeynCalc useful for multiloop calculations was to introduce some sort of topology minimization mechanism. As a typical Feynman amplitude may contain thousands of seemingly different topologies, finding mappings between those and thus reducing the number of integral families that need to be IBP-reduced is indispensable.

Here we opted for the so-called Pak algorithm [58], an approach that consists of finding one-to-one mappings between topologies by comparing a particular combination of their Symanzik polynomials $\mathcal{U}$ and $\mathcal{F}$. Normally, when switching to the Feynman parametric representation from the propagator representation, the invariance of the integral under loop momentum shifts is translated into the invariance under renamings of the Feynman parameters x_i. Pak algorithm introduces a procedure to determine a unique ordering of x_i for the given characteristic polynomial $\mathcal{P} = f(\mathcal{U}, \mathcal{F})$. Thus, integral families or single loop integrals can be conveniently compared with each other by calculating their $\mathcal{P}$ and ordering it according to Pak.

To that aim we introduced the routine `FCFeynmanPrepare` that determines the Symanzik polynomials of the given integral or topology. As far as the symbolic representation of the latter is concerned, loop integrals are called `GLI`, while topologies are represented using `FCTopology` containers. These three building blocks constitute the essence of FeynCalc's multiloop functionality.

The two main high level functions on top of that are called `FCLoopFindTopologyMappings` and `FCLoopFindIntegralMappings`. Given an input in form of `FCTopologys` or `GLIs` they can automatically work out all one-one-to-one mappings detectable by means of the Pak algorithm. In addition to that, there are many further routines for manipulating input expressions containing `GLI` and/or `FCTopology` symbols. For further information we refer to the official manual available as a PDF file [3].

In general, given some multiloop amplitude $i\mathcal{M}$ the stages of calculating it with FeynCalc will look as follows

1. Simplify $i\mathcal{M}$ using `DiracSimpliy`, `SUNSimplify` etc.
2. Identify the occurring topologies with `FCLoopFindTopologies`
   - In the case of an overdetermined set of propagators use `FCLoopCreatePartial-FractioningRules`
   - If the set of propagators is incomplete, employ `FCLoopBasisFindCompletion`
3. To map smaller topologies onto bigger ones first find all nonvanishing subtopologies via `FCLoopFindSubtopologies`
4. Minimize the number of the topologies using `FCLoopFindTopologyMappings`

[3] https://github.com/FeynCalc/feyncalc-manual/releases/tag/dev-manual

5. Apply the mappings to $i\mathcal{M}$ and eliminate all integrals in favor of `GLI`s using `FCLoopApplyTopologyMappings`

6. Do the IBP reduction for the occurring `GLI`s using external tools (e. g. FIRE or KIRA)

7. Insert the reduction tables into $i\mathcal{M}$

8. Check for one-to-one mappings between master integrals with `FCLoopFindIntegralMappings`

9. Insert analytic or numerical results for the master integrals

Although the complexity of a typical multiloop calculations might be too high for the capabilities of MATHEMATICA, in some cases FEYNCALC and FEYNHELPERS alone could be still sufficient to obtain the final result within a reasonable time frame. Moreover, the multiloop related routines can be of course also used in other contexts, e.g. when doing most of the calculation in FORM but still employing FEYNCALC at some intermediate steps.

Many ideas behind this implementation were adopted from the thesis of Jens Hoff [59] that contains a very detailed description of Pak's algorithm and related ideas. Also his unfinished MATHEMATICA package `TopoID`[4] was enormously useful for our purposes. The main algorithm of FCFEYNMANPREPARE was taken from FIRE's `FindRules` routine.

3. FeynHelpers

The main motivation behind the development of FEYNHELPERS [60] was the following observation. Even though the multiloop community is very prolific in terms of software tools for automatizing different aspects of higher order perturbative calculations, the practical applications of such codes in real-life projects is not entirely straightforward. One of the reasons is that formats of configuration files as well as input and output expressions vary from tool to tool. Using the output of one code as an input for another code always requires some conversion steps that are tedious to implement and require a formidable amount of *glue scripts* written in BASH, PYTHON, MATHEMATICA or other suitable languages. Owing to the popularity of FEYNCALC among particle physics practitioners we deemed that the format used in this package (in particular with the new `GLI` and `FCTopology` symbols) can be used as a common denominator for exchanging results between various programs.

FEYNHELPERS is implemented as a collection of interfaces between FEYNCALC and some selected tools that are commonly used in multiloop calculations. As of now we support QGRAF, PACKAGE-X [61, 62], LOOPTOOLS [18], FIRE, KIRA, FIESTA, PYSECDEC and FERMAT. The add-on provides high-level functions that accept input in FEYNCALC format and allow processing it either by directly calling the corresponding tool in the background or by generating scripts for running that tool either locally or on a cluster. Options can be used to steer the evaluation process or to adjust tool's settings. For further information we refer to the official manual available as a PDF file [5].

[4] https://github.com/thejensemann/TopoID

[5] https://github.com/FeynCalc/feynhelpers-manual/releases/tag/dev-manual

To be more specific here, let us explain how FeynHelpers can be used to perform an IBP-reduction of loop integrals that were obtained in some calculation and then loaded into FeynCalc as a list of `GLI`s with the corresponding FCTopologys.

In the first step we need to supply the topologies to the function `FIREPrepareStartFile`. This routine will generate Mathematica scripts for analyzing the given topologies using LiteRed [63] (one script per topology) and for creating `.start` or `.sbases` and `.lbases` files needed for the reduction. Depending on the complexity of the topology running those scripts can take a significant amount of time and should be in general done on a cluster. However, for simple cases where the whole process only takes few seconds, it is also possible to run them directly from an evaluation notebook by calling `FIRECreateStartFile`.

Then, we also need to create the `.config` file for the reduction as well as the list of integrals being reduced. This can be handled via `FIRECreateConfigFile` and `FIRECreateIntegralFile` respectively. The reduction itself should be definitely done on a cluster, but again as a matter of convenience for very simple cases we also offer a function `FIRERunReduction` that will start it as a background process directly from a Mathematica notebook. Finally, using `FIREImportResults` we can load the reduction tables into our notebook and convert them into a list of replacements rules where `GLI`s are substituted by a linear combination of simpler master integrals (also in the `GLI` format).

We would like to stress that this interface can be useful also in FORM-based setups where one needs to prepare FIRE runcards for a large number of integral families obtained in the course of some calculation completely unrelated to FeynCalc. The only thing one needs to do is to convert the occurring integral families and loop integrals into the `FCTopology` and `GLI` formats respectively.

4. Structure of soft-overlap contribution to $B_c \to \eta_c$ form factors

The presented tools have already been employed in a real-life multiloop calculation, where we were interested in obtaining a better understanding of QCD factorization with a systematic inclusion of power corrections. Although power corrections can be studied in the framework of SCET, some effects appearing at subleading power, in particular the end-point divergent convolution integrals, still remain problematic. As has been shown recently [64], the possible remedy in form of refactorization [65, 66] does not solve all problems in hard-exclusive processes. To illustrate this point more explicitly, it is useful to consider the $B_c \to \eta_c$ form factors in the nonrelativistic approximation ($m_b \gg m_c \gg \Lambda_{\mathrm{QCD}}$). This process can serve as a perfect laboratory to study the all-order structure of the associated double-log corrections. In particular, the all-order double-log structure at large recoil can be predicted from solving coupled integral equations and then explicitly verified using a method-of-regions analysis [67].

However, in order to check this conjecture at fixed order up to three-loops one extra ingredient is needed: the purely hard-collinear coefficient $F_{\mathrm{hc}}(\gamma)$, with

$$F(\gamma) \equiv \frac{1}{2E_\eta} \langle \eta_c(p_\eta) | \bar{c}\Gamma b | B_c(p_B) \rangle \,. \tag{1}$$

The building block $F_{\mathrm{hc}}(\gamma)$ has to be explicitly extracted from the corresponding diagrams evaluated at two and three loops, where multiloop techniques become indispensable. To be more specific, we need to consider two and three-loop QCD corrections to the process (cf. Figure 1)

$$b(m_b v^\mu)\, \bar{q}_u(m_{\bar{q}_u} v^\mu) \to W(q_1)\, q(m_q v'^\mu)\, \bar{q}_u(m_q v'^\mu), \tag{2}$$

where $\bar{q}_u$ denotes an up-type antiquark, while q stands for some other light quark. In a physical η_c meson one obviously has $\bar{q}_u = \bar{c}$ and $q = c$, but here for pedagogical purposes we choose to treat them as different species. The kinematics is chosen such that

$$v^\mu = \frac{n^\mu + \bar{n}^\mu}{2}, \quad v^2 = 1, \tag{3}$$

$$v'^\mu = \gamma n^\mu + \frac{\bar{n}^\mu}{4\gamma}, \quad (v')^2 = 1, \tag{4}$$

$$\gamma \equiv v \cdot v', \tag{5}$$

where n and $\bar{n}$ are two light-like reference vectors that satisfy

$$n^2 = \bar{n}^2 = 0, \quad n \cdot \bar{n} = 2, \tag{6}$$

so that every four-vector k can be decomposed into

$$k^\mu = \frac{\bar{n}^\mu}{2}(k \cdot n) + \frac{n^\mu}{2}(k \cdot \bar{n}) + k_\perp^\mu \equiv k_+^\mu + k_-^\mu + k_\perp^\mu. \tag{7}$$

To extract the limit we are interested in, all loop momenta k_i are considered to be hard-collinear, i. e. their components scale as $(k_+, k_-, k_\perp) \sim (\lambda^2, 1, \lambda)$. The scaling of the remaining quantities is as follows

$$m_\eta \sim \lambda^2, \quad \gamma \sim \frac{1}{\lambda^2}, \quad m_b \sim 1, \quad k_i \cdot n \sim \lambda^2, \quad k_i \cdot \bar{n} \sim 1, \quad k_i^2 \sim \lambda^2. \tag{8}$$

To ensure the correct scaling of the b-quark propagators, it is crucial, that the hard momentum $p_b = m_b v^\mu$ is always routed through the internal b-lines and the external W-line. Letting it flow through gluon or other light quark propagators would spoil the power counting and lead to wrong results.

To decrease the number of scales entering the calculation it is useful to introduce the dimensionless mass ratios

$$\bar{u}_0 \equiv \frac{m_{\bar{q}_u}}{m_\eta}, \quad u_0 \equiv \frac{m_q}{m_\eta} = 1 - \bar{u}_0, \tag{9}$$

where a physical η_c would have

$$u_0 = \bar{u}_0 = 1/2. \tag{10}$$

This way we can eliminate the light quark masses using

$$m_q = (1 - \bar{u}_0) m_\eta, \quad m_{\bar{q}_u} = \bar{u}_0 m_\eta, \tag{11}$$

so that the final result will depend only on $\bar{u}_0$, m_η, m_b and γ. Luckily, for our purposes we only need the coefficient in front of the leading pole of the amplitude, which reduces the number of scales even further. Effectively, only the $\bar{u}_0$-dependence of master integrals is nontrivial at leading power.

We perform this calculation using an automatized setup that employs FORM and FeynCalc. The code itself is called LoopScalla and will be published in near future. A preliminary version thereof is already available online[6]. In this setup we use QGRAF to generate the required diagrams

[6] https://github.com/FeynCalc/LoopScalla

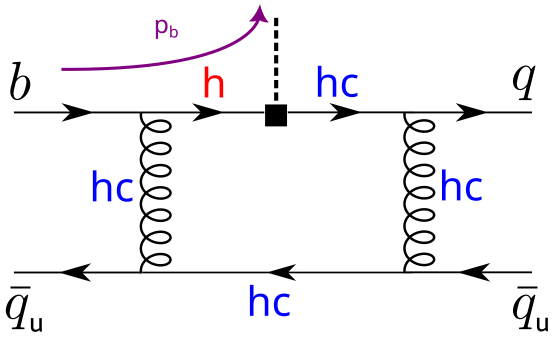

Figure 1: A representative one-loop diagram for the $B_c \to \eta_c$ transition at partonic level. The dashed line represents an emitted W-boson, while "h" and "hc" mean, that the corresponding propagators are hard or hard-collinear respectively. The purple line shows the correct routing of the b-quark momentum that respects the assigned power-counting rules. Dressing this diagram with more gluons generates higher order QCD corrections to this process.

and FORM for the insertion of Feynman rules. Dirac and color algebra simplifications as well as the expansion of the amplitudes in the hard-collinear limit are also done using FORM. Having extracted all naive topologies appearing in the amplitude we switch to FeynCalc for the purpose of finding mappings between different integral families, performing partial fractioning for cases with overdetermined propagator bases and adding extra propagators when a basis is incomplete. In addition to that, FeynCalc also generates rules for rewriting loop momentum-dependent scalar products in terms of inverse propagator denominators.

The results of this calculational step are exported as FORM `id`-statements and the insertion of the topology mappings into preliminary results is done in FORM. Then, after having extracted the final list of loop integrals for every integral family we use FeynHelpers to generate run cards for FIRE. Upon performing the IBP reduction we again employ FeynHelpers to load the reduction tables and to export them as FORM `id`-statements. These reduction rules are then converted into FORM tablebases and finally inserted into the amplitudes.

Then, all master integrals are evaluated numerically using pySecDec in order to determine the leading power of the ε-pole in each of them. Substituting those results into the final expression significantly decreases the number of integrals that need to be calculated numerically, since many masters do not contribute to the final result. Instead of calculating the remaining masters analytically, we choose a different semi-numerical approach that exploits the fact that we only need their leading poles[7]. Making a rational function ansatz

$$\sum_{i=-|a|}^{b} c_i \bar{u}_0^i + \frac{1}{1-\bar{u}_0} \sum_{i=-|a'|}^{b'} c'_i \bar{u}_0^i + \frac{1}{(1-\bar{u}_0)^2} \sum_{i=-|a''|}^{b''} c''_i \bar{u}_0^i, \quad a, b, a', b', a'', b'' = 3, 4, 5 \tag{12}$$

for each leading pole coefficient we evaluate each integral numerically at 22 special points

$$\bar{u}_0 = 1/2, 1/3, 1/4, \ldots 1/9, 2/3, 2/5, \ldots 3/4, 3/5, \ldots \tag{13}$$

[7]At three loops also subleading poles of some master integrals can enter the $1/\varepsilon^6$-piece of the full amplitude. However, at two loops the presented approach was fully sufficient.

and convert the results into rational numbers using MATHEMATICA's `Rationalize` function. Empirically, we found that numerators and denominators containing prime factors larger than 9 lead to "bad" points that should be discarded, while the remaining "good" points are kept. For example, while $5/14$ means that we have a "good" point, a point generating $113/167$ gets removed.

Putting these results together, we can generate a system of linear equations for each master integral and successfully determine all c_i, c'_i and c''_i coefficients analytically. These results can be easily cross checked by performing more evaluations using PYSECDEC or calculating some of the simpler integrals analytically. At two loops the final result for the leading pole reads

$$i\mathcal{M}^{(2)} \sim \frac{1}{\varepsilon^4}\left(C_F^2 \frac{15\bar{u}_0 + 17}{\bar{u}_0^3} - C_A C_F \frac{\bar{u}_0 + 5}{2\bar{u}_0^3}\right), \tag{14}$$

which confirms the diagrammatic analysis done in ref. [67].

The three-loop calculation is currently under way. As compared to the two-loop case, here we have to deal with 20759 diagrams (722 at two-loops) and 6276 integral families (377 at two-loops). The final results are expected to appear this year.

5. Summary

In this talk we highlighted key features of FEYNCALC 10 that implements the long-awaited routines for semi-automatic multiloop calculations. We also discussed the issue of interfacing different loop-related codes with each other and presented our solution in form of an easy-to-use FEYNCALC add-on that tackles this task for some popular programs. Last but not least, we showed a practical application of this tools to a SCET-related problem using our FORM-based framework that employs FEYNCALC and FEYNHELPERS.

Acknowledgments

The author would like to thank Rolf Mertig and Frederik Orellana for the longstanding collaboration on the FEYNCALC project. He is also grateful to Guido Bell, Philipp Böer, Thorsten Feldmann and Dennis Horstmann for the given opportunity to contribute to the study of QCD power corrections at subleading power.

Most of the calculations described in this work were done using the TTP/ITP high performance cluster at KIT and the OMNI cluster at the University of Siegen. The author would like to express his sincere gratitude to Manuel Egner, Martin Lang and Oliver Witzel for their help with the usage of the respective clusters.

This research was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 – TRR 257 "Particle Physics Phenomenology after the Higgs Discovery". This report has been assigned preprint numbers SI-HEP-2024-16 and P3H-24-043.

References

[1] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, *JHEP* **07** (2014) 079 [1405.0301].

[2] GoSam collaboration, *Automated One-Loop Calculations with GoSam*, *Eur. Phys. J. C* **72** (2012) 1889 [1111.2034].

[3] GoSam collaboration, *GOSAM-2.0: a tool for automated one-loop calculations within the Standard Model and beyond*, *Eur. Phys. J. C* **74** (2014) 3001 [1404.7096].

[4] M. Bahr et al., *Herwig++ Physics and Manual*, *Eur. Phys. J. C* **58** (2008) 639 [0803.0883].

[5] J. Bellm et al., *Herwig 7.0/Herwig++ 3.0 release note*, *Eur. Phys. J. C* **76** (2016) 196 [1512.01178].

[6] G. Bevilacqua, M. Czakon, M.V. Garzelli, A. van Hameren, A. Kardos, C.G. Papadopoulos et al., *HELAC-NLO*, *Comput. Phys. Commun.* **184** (2013) 986 [1110.1499].

[7] P. Nason, *A New method for combining NLO QCD with shower Monte Carlo algorithms*, *JHEP* **11** (2004) 040 [hep-ph/0409146].

[8] S. Frixione, P. Nason and C. Oleari, *Matching NLO QCD computations with Parton Shower simulations: the POWHEG method*, *JHEP* **11** (2007) 070 [0709.2092].

[9] S. Alioli, P. Nason, C. Oleari and E. Re, *A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX*, *JHEP* **06** (2010) 043 [1002.2581].

[10] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert et al., *Event generation with SHERPA 1.1*, *JHEP* **02** (2009) 007 [0811.4622].

[11] Sherpa collaboration, *Event Generation with Sherpa 2.2*, *SciPost Phys.* **7** (2019) 034 [1905.09127].

[12] M. Moretti, T. Ohl and J. Reuter, *O'Mega: An Optimizing matrix element generator*, hep-ph/0102195.

[13] W. Kilian, T. Ohl and J. Reuter, *WHIZARD: Simulating Multi-Particle Processes at LHC and ILC*, *Eur. Phys. J. C* **71** (2011) 1742 [0708.4233].

[14] A. Belyaev, N.D. Christensen and A. Pukhov, *CalcHEP 3.4 for collider physics within and beyond the Standard Model*, *Comput. Phys. Commun.* **184** (2013) 1729 [1207.6082].

[15] CompHEP collaboration, *CompHEP 4.4: Automatic computations from Lagrangians to events*, *Nucl. Instrum. Meth. A* **534** (2004) 250 [hep-ph/0403113].

[16] F. Yuasa et al., *Automatic computation of cross-sections in HEP: Status of GRACE system*, *Prog. Theor. Phys. Suppl.* **138** (2000) 18 [hep-ph/0007053].

[17] J. Fujimoto et al., *GRACE/SUSY automatic generation of tree amplitudes in the minimal supersymmetric standard model*, *Comput. Phys. Commun.* **153** (2003) 106 [hep-ph/0208036].

[18] T. Hahn and M. Perez-Victoria, *Automatized one loop calculations in four-dimensions and D-dimensions*, *Comput. Phys. Commun.* **118** (1999) 153 [hep-ph/9807565].

[19] J.M. Campbell et al., *Event Generators for High-Energy Physics Experiments*, *SciPost Phys.* **16** (2024) 130 [2203.11110].

[20] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, M. Kraus, B. Page et al., *Caravel: A C++ framework for the computation of multi-loop amplitudes with numerical unitarity*, *Comput. Phys. Commun.* **267** (2021) 108069 [2009.11957].

[21] G. Heinrich, S.P. Jones, M. Kerner, V. Magerya, A. Olsson and J. Schlenk, *Numerical scattering amplitudes with pySecDec*, *Comput. Phys. Commun.* **295** (2024) 108956 [2305.19768].

[22] S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M. Kerner, J. Schlenk et al., *Numerical multi-loop calculations: tools and applications*, *J. Phys. Conf. Ser.* **762** (2016) 012073 [1604.00267].

[23] S. Borowka, N. Greiner, G. Heinrich, S.P. Jones, M. Kerner, J. Schlenk et al., *Higgs Boson Pair Production in Gluon Fusion at Next-to-Leading Order with Full Top-Quark Mass Dependence*, *Phys. Rev. Lett.* **117** (2016) 012001 [1604.06447].

[24] S. Pozzorini, N. Schär and M.F. Zoller, *Two-loop tensor integral coefficients in OpenLoops*, *JHEP* **05** (2022) 161 [2201.11615].

[25] M.F. Zoller, S. Pozzorini and N. Schaer, *Towards two-loop automation in OpenLoops*, *PoS* **LL2022** (2022) 073 [2207.07468].

[26] D. Canko, G. Bevilacqua and C. Papadopoulos, *Two-Loop Amplitude Reduction with HELAC*, *PoS* **RADCOR2023** (2024) 081 [2309.14886].

[27] J.A.M. Vermaseren, *New features of FORM*, math-ph/0010025.

[28] J. Kuipers, T. Ueda, J.A.M. Vermaseren and J. Vollinga, *FORM version 4.0*, *Comput. Phys. Commun.* **184** (2013) 1453 [1203.6543].

[29] R. Harlander, T. Seidensticker and M. Steinhauser, *Complete corrections of Order alpha alpha-s to the decay of the Z boson into bottom quarks*, *Phys. Lett. B* **426** (1998) 125 [hep-ph/9712228].

[30] T. Seidensticker, *Automatic application of successive asymptotic expansions of Feynman diagrams*, in *6th International Workshop on New Computing Techniques in Physics Research: Software Engineering, Artificial Intelligence Neural Nets, Genetic Algorithms, Symbolic Algebra, Automatic Calculation*, 5, 1999 [hep-ph/9905298].

[31] M. Gerlach, F. Herren and M. Lang, *tapir: A tool for topologies, amplitudes, partial fraction decomposition and input for reductions*, *Comput. Phys. Commun.* **282** (2023) 108544 [2201.05618].

[32] Q.-f. Wu and Z. Li, *FeAmGen.jl: A Julia program for Feynman Amplitude Generation*, Comput. Phys. Commun. **301** (2024) 109230 [2310.07634].

[33] F. Feng, Y.-F. Xie, Q.-C. Zhou and S.-R. Tang, *HepLib: A C++ library for high energy physics*, Comput. Phys. Commun. **265** (2021) 107982 [2103.08507].

[34] F. Feng, S.-R. Tang and Y.-D. Gao, *HepLib: A C++ library for high energy physics (version 1.1)*, Comput. Phys. Commun. **285** (2023) 108631.

[35] J. Brod, L. Hüdepohl, E. Stamou and T. Steudtner, *MaRTIn – Manual for the "Massive Recursive Tensor Integration"*, 2401.04033.

[36] R. Mertig, M. Bohm and A. Denner, *FEYN CALC: Computer algebraic calculation of Feynman amplitudes*, Comput. Phys. Commun. **64** (1991) 345.

[37] V. Shtabovenko, R. Mertig and F. Orellana, *New Developments in FeynCalc 9.0*, Comput. Phys. Commun. **207** (2016) 432 [1601.01167].

[38] V. Shtabovenko, R. Mertig and F. Orellana, *FeynCalc 9.3: New features and improvements*, Comput. Phys. Commun. **256** (2020) 107478 [2001.04407].

[39] V. Shtabovenko, *FeynCalc goes multiloop*, J. Phys. Conf. Ser. **2438** (2023) 012140 [2112.14132].

[40] V. Shtabovenko, R. Mertig and F. Orellana, *FeynCalc 10: Do multiloop integrals dream of computer codes?*, 2312.14089.

[41] P. Nogueira, *Automatic Feynman Graph Generation*, J. Comput. Phys. **105** (1993) 279.

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

[43] A.V. Smirnov and F.S. Chuharev, *FIRE6: Feynman Integral REduction with Modular Arithmetic*, Comput. Phys. Commun. **247Â** (2020) 106877 [1901.07808].

[44] A.V. Smirnov and M. Zeng, *FIRE 6.5: Feynman integral reduction with new simplification library*, Comput. Phys. Commun. **302** (2024) 109261 [2311.02370].

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

[46] P. Maierhöfer and J. Usovitsch, *Kira 1.2 Release Notes*, 1812.01491.

[47] P. Maierhöfer and J. Usovitsch, *Recent developments in Kira*, CERN Yellow Reports: Monographs **3** (2020) 201.

[48] 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].

PoS(LL2024)071

[49] F. Lange, P. Maierhöfer and J. Usovitsch, *Developments since Kira 2.0*, *SciPost Phys. Proc.* **7** (2022) 017 [2111.01045].

[50] A.V. Smirnov, *FIESTA4: Optimized Feynman integral calculations with GPU support*, *Comput. Phys. Commun.* **204** (2016) 189 [1511.03614].

[51] A.V. Smirnov, N.D. Shapurov and L.I. Vysotsky, *FIESTA5: Numerical high-performance Feynman integral evaluation*, *Comput. Phys. Commun.* **277** (2022) 108386 [2110.11660].

[52] S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M. Kerner, J. Schlenk et al., *pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*, *Comput. Phys. Commun.* **222** (2018) 313 [1703.09692].

[53] R. Lewis, *FERMAT*, `https://home.bway.net/lewis` .

[54] C.W. Bauer, S. Fleming, D. Pirjol and I.W. Stewart, *An Effective field theory for collinear and soft gluons: Heavy to light decays*, *Phys. Rev. D* **63** (2001) 114020 [hep-ph/0011336].

[55] C.W. Bauer, D. Pirjol and I.W. Stewart, *Soft collinear factorization in effective field theory*, *Phys. Rev. D* **65** (2002) 054022 [hep-ph/0109045].

[56] M. Beneke, A.P. Chapovsky, M. Diehl and T. Feldmann, *Soft collinear effective theory and heavy to light currents beyond leading power*, *Nucl. Phys. B* **643** (2002) 431 [hep-ph/0206152].

[57] M. Beneke and T. Feldmann, *Multipole expanded soft collinear effective theory with nonAbelian gauge symmetry*, *Phys. Lett. B* **553** (2003) 267 [hep-ph/0211358].

[58] A. Pak, *The Toolbox of modern multi-loop calculations: novel analytic and semi-analytic techniques*, *J. Phys. Conf. Ser.* **368** (2012) 012049 [1111.0868].

[59] J.S. Hoff, *Methods for multiloop calculations and Higgs boson production at the LHC*, Ph.D. thesis, KIT, Karlsruhe, 2015. 10.5445/IR/1000047447.

[60] V. Shtabovenko, *FeynHelpers: Connecting FeynCalc to FIRE and Package-X*, *Comput. Phys. Commun.* **218** (2017) 48 [1611.06793].

[61] H.H. Patel, *Package-X: A Mathematica package for the analytic calculation of one-loop integrals*, *Comput. Phys. Commun.* **197** (2015) 276 [1503.01469].

[62] H.H. Patel, *Package-X 2.0: A Mathematica package for the analytic calculation of one-loop integrals*, *Comput. Phys. Commun.* **218** (2017) 66 [1612.00009].

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

[64] G. Bell, P. Böer and T. Feldmann, *Muon-electron backward scattering: a prime example for endpoint singularities in SCET*, *JHEP* **09** (2022) 183 [2205.06021].

[65] P. Böer, *QCD Factorisation in Exclusive Semileptonic B Decays New Applications and Resummation of Rapidity Logarithms*, Ph.D. thesis, Siegen U., 2018.

[66] Z.L. Liu and M. Neubert, *Factorization at subleading power and endpoint-divergent convolutions in $h \to \gamma\gamma$ decay*, *JHEP* **04** (2020) 033 [1912.08818].

[67] P. Böer, G. Bell, T. Feldmann, D. Horstmann and V. Shtabovenko, *Soft-overlap contribution to $\boldsymbol{Bc} \to \boldsymbol{\eta c}$ form factors: diagrammatic resummation of double logarithms*, *PoS* **RADCOR2023** (2024) 086 [2309.08410].