PROCEEDINGS OF SCIENCE

Two-Loop Amplitude Reduction with HELAC

Giuseppe Bevilacqua,[a] Dhimiter Canko[a,b,*] and Costas Papadopoulos[a]

[a] *Institute of Nuclear and Particle Physics, NCSR "Demokritos",*
Agia Paraskevi 15310, Greece

[b] *Department of Physics, University of Athens,*
Zografou 15784, Greece

E-mail: bevilacqua@inp.demokritos.gr, jimcanko@phys.uoa.gr, costas.papadopoulos@cern.ch

We discuss recent progress towards extending the HELAC framework to the calculation of two-loop amplitudes. A general algorithm for the automated computation of two-loop integrands is described. The algorithm covers all the steps of the computation, from the generation of loop topologies up to the construction of recursion relations for two-loop integrands. Finally, first steps towards the formulation of a new approach for reducing two-loop amplitudes to a basis of master integrals are discussed.

16th International Symposium on Radiative Corrections: Applications of Quantum Field Theory to Phenomenology (RADCOR2023)
28th May - 2nd June, 2023
Crieff, Scotland, UK

[*] Speaker

PoS(RADCOR2023)081

1. Introduction

Continuous improvements in the precision of the data collected at the LHC are demanding corresponding improvements of theoretical predictions. This is imperative for QCD processes in particular, given that hadronic interactions are ubiquitous at the LHC. The calculation of hard-scattering cross sections is at the core of perturbative QCD and is organised in terms of an expansion in powers of the strong coupling constant. On a general ground, the precision frontier of perturbative calculations to date is set to NNLO[1]. An up-to-date reference list of processes which are desired with the highest accuracy possible can be found in Ref.[1].

Generally speaking, the bottleneck of NNLO calculations is identified in double-virtual corrections, more precisely in the calculation of two-loop amplitudes. Although the workflow of two-loop calculations varies upon the specific method considered, the key aspects can be summarized as follows:

1. *construction* of two-loop integrands. This can be achieved either computing individual Feynman diagrams or using recursion relations;
2. *reduction* of two-loop amplitudes in terms of Master Integrals [2–21]. This can be achieved at integrand level or at integral level;
3. *calculation* of Master Integrals [22–40] based on analytical or (semi-)numerical methods.

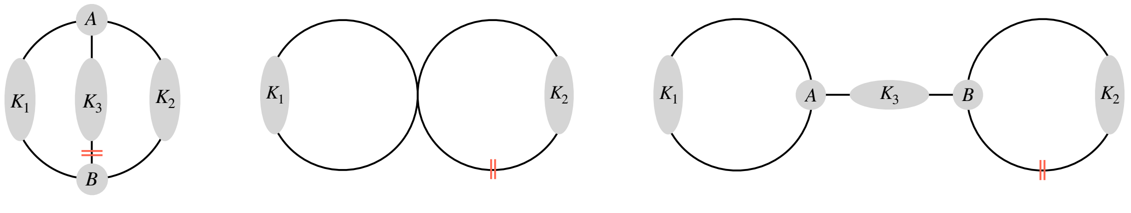
Each of these steps comes with its own challenges, and the developements during the last years have yielded a number of results for 2 → 3 cross sections at NNLO accuracy [41–45].

In this contribution we present steps towards the construction of Helac-2Loop, a framework for automated two-loop calculations. We focus in particular on items 1 and 2 in the list above. In Section 2 we illustrate the basic details of the algorithm for the computation of two-loop integrand functions. A few benchmark results are presented in Section 3. Finally, in Section 4 we sketch the approach for the reduction of two-loop amplitudes that we plan to implement in our framework.

2. Construction of two-loop integrands

Before starting the discussion, it is convenient to introduce some basic notation. We denote a generic scattering process described by n external particles with the flavors of the latter, $\{f_1, f_2, \ldots, f_n\}$. Color degrees of freedom are treated in the *color-flow* representation [46]. In this representation external gluons are labelled with a doublet of color indices $\{i, j\}$, while external quarks and anti-quarks receive the labels $\{i, 0\}$ and $\{0, j\}$ respectively. A generic color state is uniquely identified by the product $\delta^{i_{\sigma_1}}_{j_1} \delta^{i_{\sigma_2}}_{j_2} \cdots \delta^{i_{\sigma_n}}_{j_n}$, where $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ denotes a permutation of the set $\{1, 2, \ldots, n\}$. Thus, for a generic process consisting of n_g external gluons and $n_q + n_{\bar{q}}$ quarks, the number of color states is $(n_g + (n_q + n_{\bar{q}})/2)!$ (see *e.g.* Ref. [47, 48] for more details). The calculation of scattering amplitudes is organised recursively. The building blocks of the recursion are objects named currents, *i.e.* tree-level sub-amplitudes built out of a subset $S_B \subset \{1, \ldots, n\}$ of the external particles. Currents are combined into sub-amplitudes of increasing complexity till the

[1] For 2 → 2 processes the current frontier is N^{3}LO.



(a) (b) (c)

Figure 1: Master categories of two-loop topologies: *Theta* (a), *Infinity* (b), *Dumbbell* (c). The red lines indicate the propagator that is cut in order to express the two-loop topology as an equivalent $(n + 2)$-particle process at one-loop (explained in the text).

full amplitude is reached (see Ref. [47, 48] for more details). Each current (hereafter also denoted *blob*, or B) is labeled with a unique integer $\mathrm{ID(B)} = \sum_{\mathrm{i}\in\mathrm{S_B}} 2^{\mathrm{i}-1}$.

The first step towards the construction of two-loop integrands consists in the generation of *loop topologies*. A key observation is that all two-loop topologies describing arbitrary processes in the Standard Model[2] (SM) fall into one of the three master categories shown in Figure 1, that we name "*Theta*", "*Infinity*" and "*Dumbbell*" for brevity. The shaded-gray areas K_i appearing therein, that we also refer to as *sectors*, denote schematically a set $\{B_1, \ldots, B_{L_i}\}$ of tree-level blobs inserting to the i-th propagator of the loop, where $K_i = \sum_{j=1}^{L_i} \mathrm{ID(B_j)}$. Similarly A, B denote generically a blob that can possibly insert into each of the loop vertices. The number of blobs appearing in each sector, together with the ordering in which they are displaced, defines a topology. We note, however, that there is no one-to-one correspondence between blob orderings and loop topologies: the reason is that the latter are symmetric under the operations of left/-right and up-down reversal. Furthermore, *Theta* topologies are symmetric under any permutation of the sectors K_1, K_2, K_3. Configurations of blob orderings that are left-right or up-down symmetric are identified at this stage in order to avoid double counting. For the same reason, all *Theta* topologies are generated according to the rule $K_1 > K_2 > K_3$. Further details concerning the topological generation phase can be found in Ref. [49].

The topologies generated at this stage carry no information about flavor and color of the propagators in the loop. This information is generated in the next step, referred as *color-flavor dressing* (schematically illustrated in Figure 3 in the context of a simple example). The procedure can be summarised in the following steps:

1. we cut one propagator in the two-loop topology (using the conventions shown in Figure 1). In this way we take the perspective of a one-loop topology describing a $(n + 2)$-particle process with flavors $\{f_1, f_2, \ldots, f_n, F_{n+1}, F_{n+2}\}$. We refer to the latter as the *equivalent one-loop configuration* for brevity. The flavors F_{n+1}, F_{n+2} refer to the two additional particles originated by the cut and can take any value allowed to run into the loop. This step is schematised in Figure 2. We draw blobs using two different colors to stress that they are computed using different approaches. The blue blobs denote tree-level currents computed with standard Dyson-Schwinger recursion, *i.e.* all possible sub-amplitudes compatible with the external-particle content are incorporated. By contrast, the recursion defining the red

[2] More generally, this statement is true for any model whose Feynman rules include up to four-particle vertices

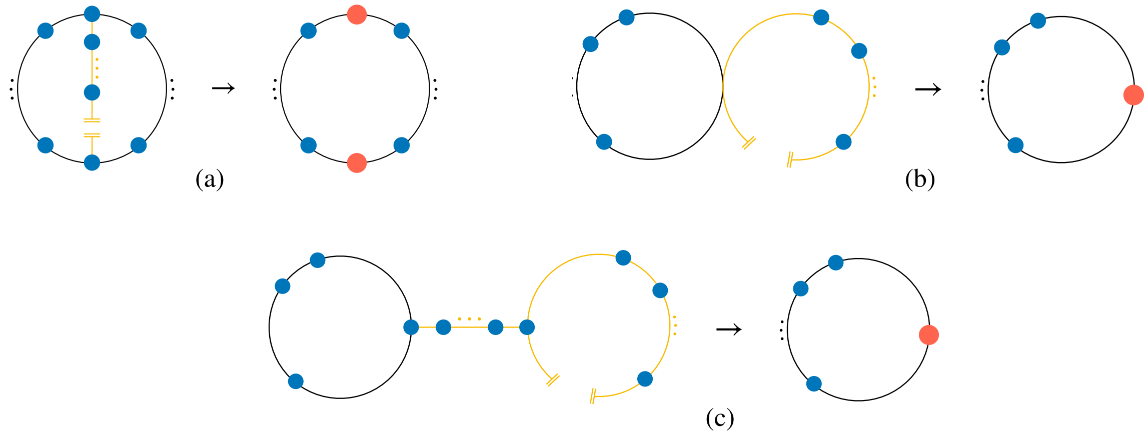


Figure 2: Relations among cut two-loop topologies and their equivalent one-loop configuration in $(n + 2)$ kinematics: *Theta* (a), *Infinity* (b), *Dumbbell* (c). The blobs denote tree-level currents which are attached to the loop propagators.

blobs is partially constrained according to the non-trivial internal structure induced by the loop propagator that has been cut (shown in yellow);

2. we treat F_{n+1}, F_{n+2} as external particles with truly independent degrees of freedom in the context of the one-loop process $\{f_1, f_2, \ldots, f_n, F_{n+1}, F_{n+2}\}$. This allows us to reuse (with proper modifications) the apparatus for one-loop integrand computations already developed in Helac-1Loop [50, 51] . Of course, F_{n+1}, F_{n+2} are constrained in both flavor and color as they are originated by cutting a propagator. Thus, in our strategy, a redundant number of degrees of freedom is considered at this intermediate stage. The latter shall be projected into the lower-dimensional space describing the physical n-particle process[3];

3. with reference to the one-loop configuration mentioned in the previous step, the construction of the integrand proceeds following the conceptual design of Helac-1Loop. The one-loop propagators are dressed with colors using all possible configurations which conserve color flow at each vertex of the loop. Similarly, propagators are dressed with flavors in all ways that are compatible with the SM. The one-loop topology, in turn, is cut, originating two additional external particles F_{n+3}, F_{n+4}. This sets the stage of a tree-level, $(n + 4)$-particle process with flavors $\{f_1, f_2, \ldots, f_n, F_{n+1}, F_{n+2}, F_{n+3}, F_{n+4}\}$. At this point the computation of one-loop numerator functions can be addressed with fully automated tree-level technology. More technical details can be found in Ref. [50, 51] .

In the last step, a set of recursion relations is constructed for each two-loop topology. These encode the instructions to compute numerators that will be used by Helac during numerical evaluations. They can be generated once and stored in a file named *skeleton*. For illustrative purposes, we provide in Figure 4 an example showing how a two-loop numerator is represented

[3] For the color degrees of freedom, the projection is understood as a contraction of the form $\left(\delta^{i_{\sigma_1}}_{j_1}\delta^{i_{\sigma_2}}_{j_2}\cdots\delta^{i_{\sigma_n}}_{j_n}\delta^{i_{\sigma_{n+1}}}_{j_{n+1}}\delta^{i_{\sigma_{n+2}}}_{j_{n+2}}\right)\left(\delta^{j_{n+1}}_{i_{n+2}}\delta^{j_{n+2}}_{i_{n+1}}\right) = (N_C)^\alpha \cdot \delta^{i_{\sigma'_1}}_{j_1}\delta^{i_{\sigma'_2}}_{j_2}\cdots\delta^{i_{\sigma'_n}}_{j_n}$, where $\{\sigma'_1, \ldots, \sigma'_n\}$ $(\{\sigma_1, \ldots, \sigma_{n+2}\})$ is a permutation of $\{1, \ldots, n\}$ $(\{1, \ldots, n+2\})$, N_C is the number of colors, and $\alpha = 0$ or 1 depending on the configuration.

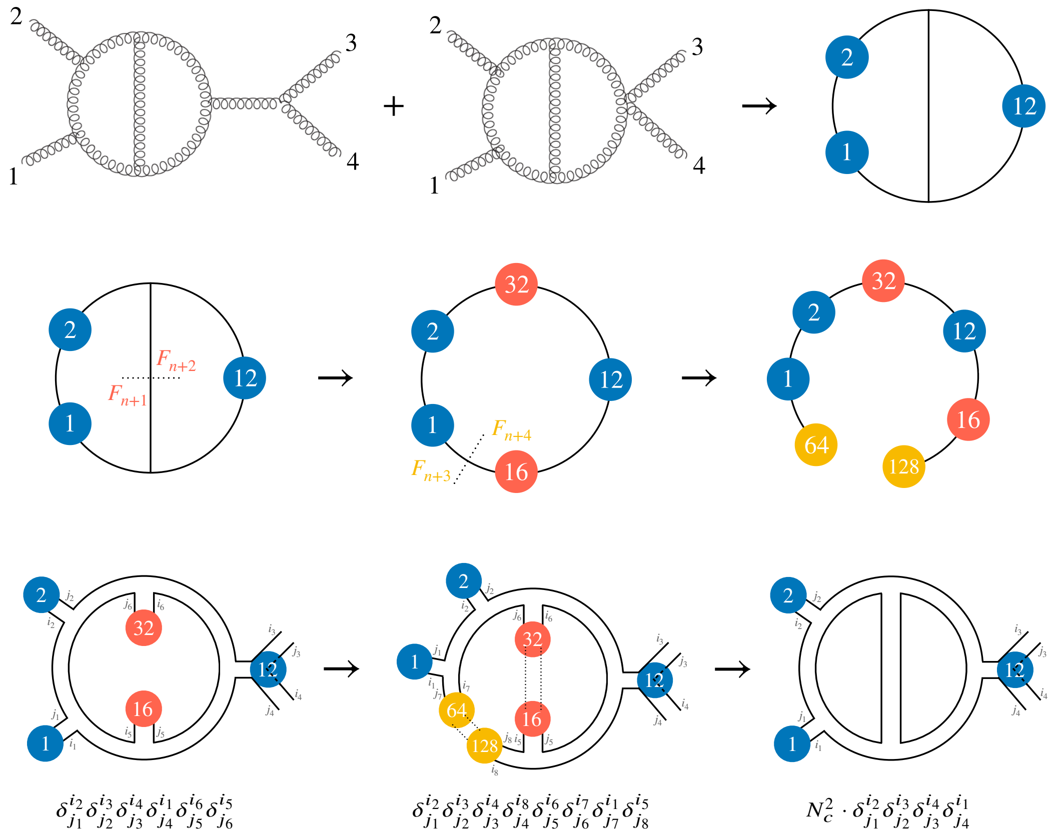

2
3
1
4
+
2
3
1
4
→
2
12
1
2
12
1
F_{n+2}
F_{n+1}
→
32
2
12
1
16
F_{n+4}
F_{n+3}
→
32
2
12
1
16
64
128
2
32
16
12
1
→
2
1
32
64
16
128
12
→
2
12
1
$\delta^{i_2}_{j_1}\delta^{i_3}_{j_2}\delta^{i_4}_{j_3}\delta^{i_1}_{j_4}\delta^{i_6}_{j_5}\delta^{i_5}_{j_6}$
$\delta^{i_2}_{j_1}\delta^{i_3}_{j_2}\delta^{i_4}_{j_3}\delta^{i_8}_{j_4}\delta^{i_6}_{j_5}\delta^{i_7}_{j_6}\delta^{i_1}_{j_7}\delta^{i_5}_{j_8}$
$N_c^2\cdot\delta^{i_2}_{j_1}\delta^{i_3}_{j_2}\delta^{i_4}_{j_3}\delta^{i_1}_{j_4}$

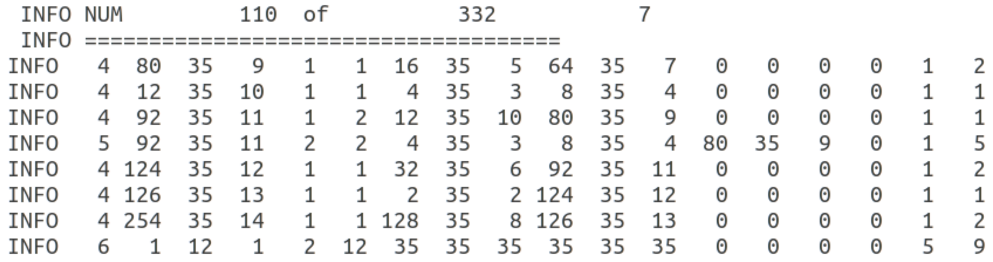
Figure 3: Schematic illustration of color-flavor dressing for a simple two-loop topology.

in our framework. The example refers to the 110th numerator out of a total of 332 numerators contributing to a specific color connection stored in the skeleton. Each line reports the instructions to compute a single current involved in the recursive calculation of the numerator. The information concerning the two-loop topology is stored in the last line.

We have performed a number of tests at the level of individual two-loop numerators in order to validate our implementation. Our results have been found in excellent agreement with the numerical output generated with the help of FeynArts + FeynCalc [52, 53].

3. Benchmark results

To give an idea of the computational cost of two-loop numerator generations, we provide in Table 1 some details concerning the requirements of CPU time and disk occupancy for a few benchmark processes. We focus in particular on four-gluon and five-gluon amplitudes. We consider the Leading Color (LC) approximation as benchmark for our comparisons. For the four-gluon case we also show results based on Full Color (FC). The purpose is to understand how the computational cost scales with increasing multiplicities and/or improved color treatment. In all cases we also report the total number of two-loop numerators as a measure of the complexity of the calculation. Comparing the first two rows of Table 1 we observe that, when going from LC to FC, the memory



```
 INFO NUM         110  of         332         7
 INFO =====================================
INFO   4  80  35   9   1   1  16  35   5  64  35   7   0   0   0   0   1   2
INFO   4  12  35  10   1   1   4  35   3   8  35   4   0   0   0   0   1   1
INFO   4  92  35  11   1   2  12  35  10  80  35   9   0   0   0   0   1   1
INFO   5  92  35  11   2   2   4  35   3   8  35   4  80  35   9   0   1   5
INFO   4 124  35  12   1   1  32  35   6  92  35  11   0   0   0   0   1   2
INFO   4 126  35  13   1   1   2  35   2 124  35  12   0   0   0   0   1   1
INFO   4 254  35  14   1   1 128  35   8 126  35  13   0   0   0   0   1   2
INFO   6   1  12   1   2  12  35  35  35  35  35  35   0   0   0   0   5   9
```

Figure 4: Internal representation of a two-loop numerator in the HELAC framework. All lines except the last one are in the same format as in HELAC-1LOOP [51]. The last line includes in addition information on the two-loop topology [54]: the second and third numbers, 1 and 12, are used to specify the denominator structure of the given numerator.

Process	N_{Loops}	Color	Numerators	Size [MB]	Timing [s]
$gg \to gg$	2	Full	89392	111	415
$gg \to gg$	2	Leading	4560	9	15
$gg \to ggg$	2	Leading	81480	300	1303
$gg \to gg$	1	Full	768	0.5	2
$gg \to ggg$	1	Full	11496	15	533

Table 1: Details of CPU time and disk space required to generate and store all two-loop numerators for a few reference processes. *Leading* refers to leading-color approximation, *Full* denotes full treatment of color degrees of freedom. All numbers have been obtained on an Intel 1.80 GHz processor with the gfortran compiler using the option -O3. Results of one-loop generations are also reported for comparison.

size increases by one order of magnitude. This is consistent with the larger number of numerators that are generated and stored in FC. Similarly the CPU time scales almost by a factor of 30. A more dramatic impact on scaling is observed, in contrast, when adding one more gluon in the computation while taking the LC viewpoint, as can be inferred by comparing the second and third lines of Table 1. In this case, CPU time and memory scale approximately by factors 90 and 30 respectively. We would like to stress that there is still room for better organisation of numerators and optimisation of the whole generation chain, which is currently under study. This is expected to improve the performance both in terms of computing efficiency and storage.

4. Steps towards a new approach for two-loop amplitude reduction

Let us consider a two-loop topology characterised by N_P loop propagators for a generic process with n external particles. Let $\{D_1, \ldots, D_{N_P}\}$ be the denominators of the loop propagators (we will refer to them as *propagators* in the following for brevity), $\{p_1, \ldots, p_n\}$ the four-dimensional momenta of the external particles and $\{k_1, k_2\}$ the loop momenta. The latter, expressed in d space-time dimensions, are decomposed as $k_i = \bar{k}_i + k_i^*$ where $\bar{k}_i$ is the four-dimensional and k_i^* the extra-dimensional part. The scalar product reads $k_i \cdot k_j = \bar{k}_i \cdot \bar{k}_j + \mu_{ij}$ $(i = 1, 2)$, where we have defined $\mu_{ij} = k_i^* \cdot k_j^*$.

The most general two-loop integrand can be constructed out of the Lorentz scalars

$$p_i \cdot p_j\,, \qquad k_i \cdot k_j\,, \qquad k_i \cdot p_j\,, \qquad k_i \cdot \eta_j\,, \tag{1}$$

where η_j are *transverse vectors* defined such that $\eta_i \cdot p_j = 0$. The integrand is understood to be a rational function of the form

$$\mathcal{R} = \frac{\mathcal{N}}{\mathcal{D}} \equiv \frac{\sum_a c_a(\vec{s}, \varepsilon) \left(z_1^{(a)}\right)^{\beta_1} \cdots \left(z_{n_a}^{(a)}\right)^{\beta_{n_a}}}{D_1 \cdots D_{N_P}}\,, \tag{2}$$

where the β's are integers, $\vec{s}$ denotes generically scalar products of the form $p_i \cdot p_j$ and $z^{(a)} \in S = \{k_i \cdot k_j,\ k_i \cdot p_j,\ k_i \cdot \eta_j\}$. Those scalars $z^{(a)}$ which admit a decomposition in terms of linear combinations of propagators D_i cancel with the denominator. Then, the integrand can be recast in the form

$$\mathcal{R} = \sum_{m=0}^{N_P} \sum_{\sigma} \frac{\sum_j \tilde{c}_j^{(\sigma)}(\vec{s}, \varepsilon) \prod_{k=1}^{n_{\mathrm{T+ISP}}^{(m)}} \left(\bar{z}_k^{(\sigma)}\right)^{\alpha_k^{(j)}}}{D_{\sigma_1} \cdots D_{\sigma_m}} \tag{3}$$

where the α's are integers and σ denotes any possible subset of $\{1, \ldots, N_P\}$ consisting of m elements. The residual scalar products appearing in the numerator (that we label as $\bar{z}_k^{(j)}$ for clarity) include both the transverse ones (T), $k_i \cdot \eta_j$, as well as what is known as *irreducible scalar products* (ISP). Based on Eq.(3), one can express the numerator function $\mathcal{N}$ using the following equation:

$$\mathcal{N} = \sum_{m=0}^{N_P} \sum_{\sigma} \sum_{j} \tilde{c}_j^{(\sigma)}(\vec{s}, \varepsilon) \prod_{k=1}^{n_{\mathrm{T+ISP}}^{(m)}} \left(\bar{z}_k^{(\sigma)}\right)^{\alpha_k^{(j)}} \prod_{i \notin \sigma} D_i\,. \tag{4}$$

Our goal is to find a method to extract numerically the coefficients $\tilde{c}_j^{(\sigma)}(\vec{s}, \varepsilon)$ at the *integrand* level. Terms in Eq.(4) which do not depend explicitly on transverse vectors (η_j) lead to Feynman integrals which can be addressed using integration-by-part (IBP) techniques [55–59]. Scalar products of the form $(k_i \cdot \eta_j)^P$ (where P is a positive integer) can be eliminated, if present, using the following considerations:

- if P is *odd*, the corresponding terms vanish upon integration;
- if P is *even*, the corresponding terms can be cast in a form which vanish upon integration using so-called *traceless completion* [2, 60]. This is equivalent to a change of basis of the form

$$(k_i \cdot \eta_j)(k_l \cdot \eta_j) \longrightarrow (k_i \cdot \eta_j)(k_l \cdot \eta_j) - \frac{\mu_{il}}{d-4} \tag{5}$$

$$(k_i \cdot \eta_j)^2 (k_l \cdot \eta_j)^2 \longrightarrow (k_i \cdot \eta_j)^2 (k_l \cdot \eta_j)^2 - \frac{(k_i \cdot \eta_j)^2 \mu_{ll} + (k_l \cdot \eta_j)^2 \mu_{ii} + 4(k_i \cdot \eta_j)(k_l \cdot \eta_j)\mu_{il}}{2(d-4)} \tag{6}$$

Following the paradigm developed for one-loop calculations, and having all the aforementioned in mind, a possible way of implementing the amplitude reduction within Helac-2Loop can be sketched in the following steps:

1. extracting the 4-dimensional part of $\tilde{c}_j^{(\sigma)}(\vec{s}, \varepsilon)$ using values for the loop-momenta obtained from the cut equations $D_{\sigma_1} = \cdots = D_{\sigma_m} = 0$ in an OPP-like approach [61, 62];

2. finding the polynomial dependence on μ_{ij} terms (analogously to the treatment of $\mathcal{R}_1$ terms in OPP reduction);

3. determining the missing ε-dimensional part of the numerator using two-loop rational terms ($\mathcal{R}_2$ terms).

4. reducing the Feynman integrals containing ISP monomials to Master Integrals using IBP reduction.

Notice that all four steps described above are universal. The integrand basis (steps 1 and 2), the $\mathcal{R}_2$ terms (step 3) and the IBP reduction (step 4) are independent of the specific process under consideration. On the other hand, we expect that the IBP reduction presents technical challenges when applied to certain two-loop topologies with 5 or more external legs. Moreover, the implementation of $\mathcal{R}_2$ rational terms (following up Ref. [63–65]) requires further study.

5. Summary

We have presented an approach for computing two-loop integrands for arbitrary processes in a fully automated way. Furthermore we have sketched a possible strategy for the reduction of two-loop amplitudes to Master Integrals. These results are part of the ongoing efforts to develop Helac-2Loop, a framework for automated two-loop amplitude calculation.

Acknowledgments

This work is supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers" (Project Number: 02674 HOCTools-II).

References

[1] A. Huss, J. Huston, S. Jones and M. Pellen, J. Phys. G **50** (2023) no.4, 043001 [arXiv:2207.02122 [hep-ph]].

[2] H. Ita, Phys. Rev. D **94** (2016) no.11, 116015 [arXiv:1510.05626 [hep-th]].

[3] P. Mastrolia, T. Peraro and A. Primo, JHEP **08** (2016), 164 [arXiv:1605.03157 [hep-ph]].

[4] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, M. Kraus, B. Page, E. Pascual, M. S. Ruf and V. Sotnikov, Comput. Phys. Commun. **267** (2021), 108069 [arXiv:2009.11957 [hep-ph]].

[5] S. Badger, H. B. Hartanto, J. Kryś and S. Zoia, JHEP **05** (2022), 035 [arXiv:2201.04075 [hep-ph]].

[6] S. Badger, H. B. Hartanto, J. Kryś and S. Zoia, JHEP **11** (2021), 012 [arXiv:2107.14733 [hep-ph]].

[7] S. Badger, H. B. Hartanto and S. Zoia, Phys. Rev. Lett. **127** (2021) no.1, 012001 [arXiv:2102.02516 [hep-ph]].

[8] H. B. Hartanto, S. Badger, C. Brønnum-Hansen and T. Peraro, JHEP **09** (2019), 119 [arXiv:1906.11862 [hep-ph]].

[9] S. Abreu, F. Febres Cordero, H. Ita, M. Klinkert, B. Page and V. Sotnikov, JHEP **04** (2022), 042 [arXiv:2110.07541 [hep-ph]].

[10] S. Badger, C. Brønnum-Hansen, D. Chicherin, T. Gehrmann, H. B. Hartanto, J. Henn, M. Marcoli, R. Moodie, T. Peraro and S. Zoia, JHEP **11** (2021), 083 [arXiv:2106.08664 [hep-ph]].

[11] H. A. Chawdhry, M. Czakon, A. Mitov and R. Poncelet, JHEP **07** (2021), 164 [arXiv:2103.04319 [hep-ph]].

[12] B. Agarwal, F. Buccioni, A. von Manteuffel and L. Tancredi, JHEP **04** (2021), 201 [arXiv:2102.01820 [hep-ph]].

[13] H. A. Chawdhry, M. Czakon, A. Mitov and R. Poncelet, JHEP **06** (2021), 150 [arXiv:2012.13553 [hep-ph]].

[14] S. Abreu, B. Page, E. Pascual and V. Sotnikov, JHEP **01** (2021), 078 [arXiv:2010.15834 [hep-ph]].

[15] S. Abreu, F. Febres Cordero, H. Ita, B. Page and V. Sotnikov, JHEP **07** (2021), 095 [arXiv:2102.13609 [hep-ph]].

[16] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, B. Page and V. Sotnikov, JHEP **05** (2019), 084 [arXiv:1904.00945 [hep-ph]].

[17] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita and B. Page, Phys. Rev. Lett. **122** (2019) no.8, 082002 [arXiv:1812.04586 [hep-ph]].

[18] S. Abreu, G. De Laurentis, H. Ita, M. Klinkert, B. Page and V. Sotnikov, [arXiv:2305.17056 [hep-ph]].

[19] B. Agarwal, F. Buccioni, A. von Manteuffel and L. Tancredi, Phys. Rev. Lett. **127** (2021) no.26, 262001 [arXiv:2105.04585 [hep-ph]].

[20] T. Peraro, JHEP **12** (2016), 030 [arXiv:1608.01902 [hep-ph]].

[21] S. Pozzorini, N. Schär and M. F. Zoller, JHEP **05** (2022), 161 [arXiv:2201.11615 [hep-ph]].

[22] A. V. Kotikov, Phys. Lett. B **254** (1991), 158-164.

[23] A. V. Kotikov, Phys. Lett. B **267** (1991), 123-127 [erratum: Phys. Lett. B **295** (1992), 409-409].

[24] T. Gehrmann and E. Remiddi, Nucl. Phys. B **580** (2000), 485-518 [arXiv:hep-ph/9912329 [hep-ph]].

[25] J. M. Henn, Phys. Rev. Lett. **110** (2013), 251601 [arXiv:1304.1806 [hep-th]].

[26] C. G. Papadopoulos, JHEP **07** (2014), 088 [arXiv:1401.6057 [hep-ph]].

[27] C. G. Papadopoulos, D. Tommasini and C. Wever, JHEP **04** (2016), 078 [arXiv:1511.09404 [hep-ph]].

[28] X. Liu, Y. Q. Ma and C. Y. Wang, Phys. Lett. B **779** (2018), 353-357 [arXiv:1711.09572 [hep-ph]].

[29] T. Gehrmann, J. M. Henn and N. A. Lo Presti, JHEP **10** (2018), 103 [arXiv:1807.09812 [hep-ph]].

[30] D. Chicherin, T. Gehrmann, J. M. Henn, P. Wasser, Y. Zhang and S. Zoia, Phys. Rev. Lett. **123** (2019) no.4, 041603 [arXiv:1812.11160 [hep-ph]].

[31] D. Chicherin and V. Sotnikov, JHEP **20** (2020), 167 [arXiv:2009.07803 [hep-ph]].

[32] S. Abreu, H. Ita, F. Moriello, B. Page, W. Tschernow and M. Zeng, JHEP **11** (2020), 117 [arXiv:2005.04195 [hep-ph]].

[33] D. D. Canko, C. G. Papadopoulos and N. Syrrakos, JHEP **01** (2021), 199 [arXiv:2009.13917 [hep-ph]].

[34] S. Abreu, H. Ita, B. Page and W. Tschernow, JHEP **03** (2022), 182 [arXiv:2107.14180 [hep-ph]].

[35] A. Kardos, C. G. Papadopoulos, A. V. Smirnov, N. Syrrakos and C. Wever, JHEP **05** (2022), 033 [arXiv:2201.07509 [hep-ph]].

[36] D. Chicherin, V. Sotnikov and S. Zoia, JHEP **01** (2022), 096 [arXiv:2110.10111 [hep-ph]].

[37] S. Abreu, D. Chicherin, H. Ita, B. Page, V. Sotnikov, W. Tschernow and S. Zoia, [arXiv:2306.15431 [hep-ph]].

[38] T. Binoth and G. Heinrich, Nucl. Phys. B **585** (2000), 741-759 [hep-ph/0004013].

[39] M. Borinsky, [arXiv:2008.12310 [math-ph]].

[40] F. Moriello, JHEP **01** (2020), 150 [arXiv:1907.13234 [hep-ph]].

[41] S. Kallweit, V. Sotnikov and M. Wiesemann, Phys. Lett. B **812** (2021), 136013 [arXiv:2010.04681 [hep-ph]].

[42] H. A. Chawdhry, M. Czakon, A. Mitov and R. Poncelet, JHEP **09** (2021), 093 [arXiv:2105.06940 [hep-ph]].

[43] M. Czakon, A. Mitov and R. Poncelet, Phys. Rev. Lett. **127** (2021) no.15, 152001 [erratum: Phys. Rev. Lett. **129** (2022) no.11, 119901] [arXiv:2106.05331 [hep-ph]].

[44] H. B. Hartanto, R. Poncelet, A. Popescu and S. Zoia, Phys. Rev. D **106** (2022) no.7, 074016 [arXiv:2205.01687 [hep-ph]].

[45] S. Badger, M. Czakon, H. B. Hartanto, R. Moodie, T. Peraro, R. Poncelet and S. Zoia, [arXiv:2304.06682 [hep-ph]].

[46] G. 't Hooft, Nucl.Phys.B 72 (1974) 461.

[47] A. Kanaki and C. G. Papadopoulos, Comput.Phys.Commun. 132 (2000) 306-315 [arXiv:hep-ph/0002082].

[48] A. Cafarella, C. G. Papadopoulos and M. Worek, Comput.Phys.Commun. 180 (2009) 1941-1955 [arXiv:0710.2427 [hep-ph]].

[49] G. Bevilacqua, D. D. Canko, A. Kardos and C. G. Papadopoulos, J. Phys. Conf. Ser. **2105** (2021) no.5, 012010.

[50] A. van Hameren, C. G. Papadopoulos and R. Pittau, JHEP **09** (2009), 106 [arXiv:0903.4665 [hep-ph]].

[51] G. Bevilacqua, M. Czakon, M. V. Garzelli, A. van Hameren, A. Kardos, C. G. Papadopoulos, R. Pittau and M. Worek, Comput. Phys. Commun. **184** (2013), 986-997 [arXiv:1110.1499 [hep-ph]].

[52] T. Hahn, Comput. Phys. Commun. **140** (2001), 418-431 [arXiv:hep-ph/0012260 [hep-ph]].

[53] V. Shtabovenko, R. Mertig and F. Orellana, Comput. Phys. Commun. **256** (2020), 107478 [arXiv:2001.04407 [hep-ph]].

[54] G. Bevilacqua, D. D. Canko and C. G. Papadopoulos, work in progress.

[55] K. G. Chetyrkin and F. V. Tkachov, Nucl. Phys. B **192** (1981), 159-204.

[56] F. V. Tkachov, Phys. Lett. B **100** (1981), 65-68.

[57] S. Laporta, Int. J. Mod. Phys. A **15** (2000), 5087-5159 [arXiv:hep-ph/0102033 [hep-ph]].

[58] A. V. Smirnov and F. S. Chuharev, Comput. Phys. Commun. **247** (2020), 106877 [arXiv:1901.07808 [hep-ph]].

[59] J. Klappert, F. Lange, P. Maierhöfer and J. Usovitsch, Comput. Phys. Commun. **266** (2021), 108024 [arXiv:2008.06494 [hep-ph]].

[60] V. Sotnikov, PhD Thesis (2019), DOI: 10.6094/UNIFR/151540.

[61] G. Ossola, C. G. Papadopoulos and R. Pittau, Nucl. Phys. B **763** (2007), 147-169 [arXiv:hep-ph/0609007 [hep-ph]].

[62] G. Ossola, C. G. Papadopoulos and R. Pittau, JHEP **03** (2008), 042 [arXiv:0711.3596 [hep-ph]].

[63] S. Pozzorini, H. Zhang and M. F. Zoller, JHEP **05** (2020), 077 [arXiv:2001.11388 [hep-ph]].

[64] J. N. Lang, S. Pozzorini, H. Zhang and M. F. Zoller, JHEP **10** (2020), 016 [arXiv:2007.03713 [hep-ph]].

[65] J. N. Lang, S. Pozzorini, H. Zhang and M. F. Zoller, JHEP **01** (2022), 105 [arXiv:2107.10288 [hep-ph]].