Vol. 44 (2013) *ACTA PHYSICA POLONICA B* No 11

INTEGRAND-LEVEL REDUCTION AT ONE AND HIGHER LOOPS*

TIZIANO PERARO

Max-Planck Insitut für Physik
Föhringer Ring 6, 80805 München, Germany

(Received October 21, 2013)

The integrand-level reduction of scattering amplitudes is a method for the decomposition of loop integrals which has already been successfully applied and automated at one-loop, and recently extended to higher loops. We present recent developments on the topic, within a coherent framework which can be applied to any integrand at any loop order. We focus on semi-analytic and algebraic techniques, such as the improved one-loop reduction via Laurent series expansion with the library Ninja, and the multi-loop divide-and-conquer approach which can always be used to algebraically find the integrand decomposition of any Feynman graph.

DOI:10.5506/APhysPolB.44.2215
PACS numbers: 11.15.Bt, 12.38.Bx

1. Introduction

Scattering amplitudes are analytic functions of the kinematic variables of the interacting particles, hence they are determined by their singularities [1, 2]. The integrand reduction methods, developed for one-loop diagrams [3, 4] and recently extended to higher loops [5–9], exploit the knowledge of the singularity structure of the integrands to decompose the (integrated) amplitudes in terms of Master Integrals (MIs). The numerator of a Feynman diagram can be expressed as a combination of (products of) denominators, multiplied by polynomials corresponding to the *residues* at the multiple cuts of the diagrams. The multiple-cut conditions, which put some of the loop-momenta on-shell, can be viewed as *projectors* isolating each residue.

In Ref. [10] we proposed a new method for the integrand reduction of one-loop amplitudes, which simplifies the evaluation of the coefficients of the MIs by performing a Laurent expansion with respect to the variables which are

* Presented at the XXXVII International Conference of Theoretical Physics "Matter to the Deepest" Ustroń, Poland, September 1–6, 2013.

not fixed by the cut conditions. The algorithm has been implemented in the semi-numerical C++ library Ninja, which proved to be faster and numerically more stable than the original integrand-reduction approach. The library has been used for the computation of NLO QCD corrections to the Higgs boson production in association with a top-quark pair and a jet [11].

In Refs. [7, 8] the determination of the residues at the multiple cuts has been formulated as a problem of *multivariate polynomial division*, and solved at any order in the perturbation theory using algebraic geometry techniques. These allow to find the most general parametric form of a residue, whose unknown coefficients can then be found by evaluating the integrand on values of the loop momenta such that some loop denominators are on-shell, as traditionally done in the one-loop case (*fit-on-the-cut approach*). Some of these coefficients are then identified with the ones which multiply the MIs. Applying the same principles one can also perform the full integrand decomposition with purely algebraic operations, within what we call the divide-and-conquer approach [9], with successive polynomial divisions, which at each step generate the actual residues. Given its wider range of applicability, we may consider the latter a more general method for the integrand decomposition of loop integrals.

2. Integrand reduction formula

An arbitrary ℓ-loop graph represents a d-dimensional integral of the form

$$\int d^d q_1 \ldots d^d q_\ell \, \mathcal{I}_{i_1 \ldots i_n} \,, \qquad \mathcal{I}_{i_1 \ldots i_n} \equiv \frac{\mathcal{N}_{i_1 \ldots i_n}}{D_{i_1} \ldots D_{i_n}} \,, \tag{1}$$

where $i_1, \ldots, i_n$ are (not necessarily distinct) indices labeling loop propagators. The numerator $\mathcal{N}$ and the denominators D_i are polynomials in a set of coordinates $\boldsymbol{z}$. Let $P[\boldsymbol{z}]$ be the ring of all polynomials in such coordinates. Every set of indices $\{i_1, \ldots, i_n\}$ defines the ideal

$$\mathcal{J}_{i_1 i_2 \ldots i_n} \equiv \langle D_{i_1}, \ldots, D_{i_n} \rangle = \left\{ \sum_{k=1}^{n} h_k(\boldsymbol{z}) \, D_{i_k}(\boldsymbol{z}) \ : \ h_k(\boldsymbol{z}) \in P[\boldsymbol{z}] \right\}. \tag{2}$$

The goal of the integrand reduction is to find a decomposition of the integrand of the form

$$\mathcal{I}_{i_1 \ldots i_n} \equiv \frac{\mathcal{N}_{i_1 \ldots i_n}}{D_{i_1} \ldots D_{i_n}} = \sum_{k=0}^{n} \sum_{\{j_1 \ldots j_k\}} \frac{\Delta_{j_1 \ldots j_k}}{D_{j_1} \ldots D_{j_k}} \,,$$

where the *residues* $\Delta_{i_1 \ldots i_n}$ are irreducible polynomials, *i.e.* polynomials which contain no contribution belonging in the corresponding ideal $\mathcal{J}_{i_1 \ldots i_n}$.

The numerator $\mathcal{N}$ of the integrand can be decomposed by performing the *multivariate polynomial division* modulo a Gröbner basis $\mathcal{G}_{i_1\ldots i_n}$ of $\mathcal{J}_{i_1\ldots i_n}$ as

$$\mathcal{N}_{i_1\ldots i_n} = \Gamma_{i_1\ldots i_n} + \Delta_{i_1\ldots i_n} = \sum_{k=1}^{n} \mathcal{N}_{i_1\ldots i_{k-1}i_{k+1}\ldots i_n} D_{i_k} + \Delta_{i_1\ldots i_n} \qquad (3)$$

in terms of a quotient $\Gamma_{i_1\ldots i_n}$ and the remainder $\Delta_{i_1\ldots i_1\,\ldots\, i_n\ldots i_n}$. The properties of Gröbner bases ensure that the remainder is irreducible, therefore it is identified with the residue of the multiple cut $D_{i_1} = \cdots = D_{i_n} = 0$, as suggested by the notation. The quotient Γ, instead, belongs to the ideal $\mathcal{J}$, thus it can be written as a combination of denominators, as we deed in the last equality of Eq. (3). Substituting Eq. (3) in Eq. (1), we obtain the recursive formula [8, 9]

$$\mathcal{I}_{i_1\ldots i_n} = \sum_{k=1}^{n} \mathcal{I}_{i_1\ldots i_{k-1}i_{k+1}\ldots i_n} + \frac{\Delta_{i_1\ldots\, i_n}}{D_{i_1}\ldots D_{i_n}}\,. \qquad (4)$$

Equation (4) expresses a given integrand in terms of an irreducible residue sitting over its denominators and a sum integrands with fewer denominators. Hence, the recursive application of this formula ultimately yields the full decomposition of any integrand in terms of irreducible residues and denominators, as in Eq. (2).

The existence of such a recursive formula proves that the integrand decomposition can be extended at any number of loops and the most general parametrization of a residue $\Delta_{i_1\ldots i_n}$ can be identified with the most general remainder of a polynomial division modulo the Gröbner basis $\mathcal{G}_{i_1\ldots i_n}$. Within the *fit-on-the-cut* approach such unknown coefficients can be found by evaluating the numerator on the solutions of the multiple cut $D_{i_1} = \cdots = D_{i_n} = 0$. This method has been traditionally used at one loop, and recently applied to higher-loop amplitudes as well [5, 6, 12–14].

In Ref. [8], we applied the recursive formula in Eq. (4) to the most general one-loop integrand. This allowed to easily derive the well know OPP decomposition for dimensionally-regulated one-loop amplitudes [3, 4], as well as its higher-rank generalization for effective and non-renormalizable theories [10] implemented in Xsamurai [15] (which extends the Samurai library [16]) and used *e.g.* in the computation of NLO QCD corrections to the Higgs boson production plus two [17] and three jets [18] in gluon fusion, in the infinite top-mass approximation.

3. Integrand-reduction via Laurent expansion with Ninja

An improved version of integrand-reduction method for one-loop amplitudes was presented in [10], elaborating on the techniques proposed in [19, 20].

Whenever the analytic dependence of the integrand on the loop momentum is known, this method allows to extract the unknown coefficients of the residues by performing a Laurent expansion with respect to one of the free components of the loop momenta which are not fixed by the on-shell conditions.

Within the original integrand reduction algorithm [16, 21, 22], the determination of the unknown coefficients requires to sample the numerator on a finite subset of the on-shell solutions, subtract from the integrand all the non-vanishing contributions coming from higher-point residues, and solve the resulting linear system of equations. Since in the asymptotic limit both the integrand and the higher-point subtractions exhibit the same polynomial behavior as the residue, with the Laurent-expansion method one can instead identify the unknown coefficients with the ones of the expansion of the integrand, corrected by the contributions coming from higher-point residues. In other words, with this approach the system of equations for the coefficients becomes diagonal and the subtractions of higher-point contributions can be implemented as *corrections at the coefficient level* which replace the subtractions at the integrand level of the original algorithm. The parametric form of this corrections can be computed once and for all, in terms of a subset of the higher-point coefficients.

This reduction algorithm has been implemented in the semi-numerical C++ library Ninja, which has been interfaced with the package GoSam [23] for automated one-loop computations. Since the integrand of a loop amplitude is a rational function, its semi-numerical Laurent expansion has been implemented as a simplified polynomial division between the numerator and the denominators.

The input of the method implemented in Ninja is the numerator cast in four different forms (one of which is optional). These can all be easily and very quickly generated from the knowledge of the analytic dependence of the integrand on the loop momentum (*e.g.* from the analytic expression of the numerator generated by GoSam). The first form corresponds to a simple evaluation of the numerator as a function of the loop momentum, while the others must return the leading terms of a parametric Laurent expansion of the numerator. Ninja computes the parametric solutions of each multiple cut, performs the Laurent expansion via a simplified polynomial division between the (expansions of the) numerator and the denominators, and implements the subtractions at the coefficient level in order to get the unknown coefficients. These are then multiplied by the corresponding MIs. Ninja implements a wrapper of the OneLoop library [24, 25] which caches the values of computed integrals and allows constant-time lookups from their arguments. The library can also be used for the reduction of higher-rank integrands where the rank of a numerator can exceed the number of denominators by one. The simplified fit of the coefficients and the subtractions at coefficient level make the algorithm implemented in Ninja significantly lighter, faster and more stable than the original.

The first new phenomenological application of Ninja has been the computation of NLO QCD corrections to the Higgs boson production in association with a top-quark pair and a jet [11]. The possibility of exploiting the improved stability of the new algorithm has been especially important for the computation of the corresponding six-point virtual amplitude, given the presence of two mass scales as well as massive loop propagators which make traditional integrand reduction algorithms numerically unstable. Indeed, for the highly non-trivial process under consideration, only a number of phase-space points of the order of one per mill were detected as unstable. All these points have been recovered using the tensorial reduction provided by Golem95 [26, 27], avoiding the need necessity of higher precision routines.

4. Divide-and-conquer approach

The direct application of the integrand reduction formula of Eq. (4) on the numerator of an l-loop graph allows to perform the integrand decomposition algebraically by successive polynomial divisions, within what we call the divide-and-conquer approach [9]. At each step, the remainders of the divisions are identified with the residues of the corresponding set of denominators, while the quotients become the numerators of the lower-point integrands appearing on the r.h.s. of the formula, allowing thus to iterate the procedure. In this way, the decomposition of any integrand is obtained analytically, with a finite number of algebraic operations, without requiring the knowledge of the varieties of solutions of the multiple cuts, nor the one of the parametric form of the residues.

This algorithm has been automated in a Python package which uses Form [28] and Macaulay2 for the algebraic operations and has been applied to the examples depicted in Fig. 1. Despite their simplicity, these show the broadness of applicability of the method which is not affected by the presence of massive propagators, non-planar diagrams, higher powers of loop denominators or higher-rank contributions in the numerator.

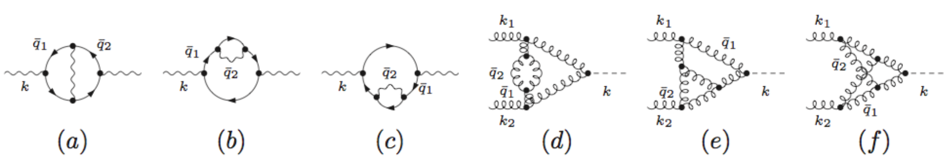

Fig. 1. Examples of diagrams reduced using the divide-and-conquer approach.

5. Conclusions

We described a coherent framework for the decomposition of Feynman integrals, which can be applied at any loop order, regardless of the complexity of the integrand, the number of external legs or the presence of higher powers of loop denominators. This framework allows to easily derive well known results at one-loop order and extend them to higher loops.

In the one-loop case, we showed how the knowledge of the analytic structure of the integrands on the multiple cuts, and in particular their asymptotic behavior on the on-shell solutions, can be used to improve the analytic reduction with the Laurent expansion method. Its implementation in the C++ library Ninja provided a considerable gain in the speed and in the stability of the reduction.

At higher loops, we described the application of the divide-and-conquer approach, which allows to perform the full decomposition with purely algebraic operations on the numerator and the set of denominators of a given integrand, without requiring the knowledge of the algebraic variety defined by the on-shell solutions. We applied it to simple examples, some of which cannot be addressed with other unitarity-based and integrand-reduction methods, due to the presence of higher powers of loop denominators in the integrands. Since it is based on the same principles used to constructively prove the existence of the integrand decomposition at all loops, the divide-and-conquer approach does not have the limitations of other methods and can be considered a more general integrand reduction algorithm.

The work presented in this paper has been done in collaboration with H. van Deurzen, G. Luisoni, P. Mastrolia, E. Mirabella and G. Ossola. This work is supported by the Alexander von Humboldt Foundation, in the framework of the Sofja Kovaleskaja Award Project "Advanced Mathematical Methods for Particle Physics", endowed by the German Federal Ministry of Education and Research.

REFERENCES

[1] Z. Bern, L.J. Dixon, D.C. Dunbar, D.A. Kosower, *Nucl. Phys.* **B425**, 217 (1994).

[2] F. Cachazo, P. Svrcek, E. Witten, *J. High Energy Phys.* **0409**, 006 (2004).

[3] G. Ossola, C.G. Papadopoulos, R. Pittau, *Nucl. Phys.* **B763**, 147 (2007).

[4] R.K. Ellis, W.T. Giele, Z. Kunszt, *J. High Energy Phys.* **0803**, 003 (2008).

[5] P. Mastrolia, G. Ossola, *J. High Energy Phys.* **1111**, 014 (2011).

[6] S. Badger, H. Frellesvig, Y. Zhang, *J. High Energy Phys.* **1204**, 055 (2012).

[7] Y. Zhang, *J. High Energy Phys.* **1209**, 042 (2012).

[8] P. Mastrolia, E. Mirabella, G. Ossola, T. Peraro, *Phys. Lett.* **B718**, 173 (2012).

[9] P. Mastrolia, E. Mirabella, G. Ossola, T. Peraro, `arXiv:1307.5832 [hep-ph]`.

[10] P. Mastrolia, E. Mirabella, T. Peraro, *J. High Energy Phys.* **1206**, 095 (2012) [*Erratum-ibid.* **1211**, 128 (2012)].

[11] H. van Deurzen *et al.*, `arXiv:1307.8437 [hep-ph]`.

[12] S. Badger, H. Frellesvig, Y. Zhang, `arXiv:1310.1051 [hep-ph]`.

[13] P. Mastrolia, E. Mirabella, G. Ossola, T. Peraro, *Phys. Rev.* **D87**, 085026 (2013).

[14] S. Badger, H. Frellesvig, Y. Zhang, *J. High Energy Phys.* **1208**, 065 (2012).

[15] H. van Deurzen, *Acta Phys. Pol. B* **44**, 2223 (2013), this issue.

[16] P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, *J. High Energy Phys.* **1008**, 080 (2010).

[17] H. van Deurzen *et al.*, *Phys. Lett.* **B721**, 74 (2013).

[18] G. Cullen *et al.*, *Phys. Rev. Lett.* **111**, 131801 (2013).

[19] D. Forde, *Phys. Rev.* **D75**, 125019 (2007).

[20] S.D. Badger, *J. High Energy Phys.* **0901**, 049 (2009).

[21] G. Ossola, C.G. Papadopoulos, R. Pittau, *J. High Energy Phys.* **0803**, 042 (2008).

[22] P. Mastrolia, G. Ossola, C.G. Papadopoulos, R. Pittau, *J. High Energy Phys.* **0806**, 030 (2008).

[23] G. Cullen *et al.*, *Eur. Phys. J.* **C72**, 1889 (2012).

[24] A. van Hameren, *Comput. Phys. Commun.* **182**, 2427 (2011).

[25] A. van Hameren, C.G. Papadopoulos, R. Pittau, *J. High Energy Phys.* **0909**, 106 (2009).

[26] T. Binoth *et al.*, *Comput. Phys. Commun.* **180**, 2317 (2009).

[27] G. Cullen *et al.*, *Comput. Phys. Commun.* **182**, 2276 (2011).

[28] J.A.M. Vermaseren, `arXiv:math-ph/0010025`.