

A recursive one-loop algorithm for many-particle amplitudes

Fabio Cascioli

University of Zürich

E-mail: cascio- @
physik.uzh.ch

Philipp Maierhöfer*

University of Zürich

E-mail: philipp@physik.uzh.ch

Stefano Pozzorini

University of Zürich

E-mail: pozzorin@physik.uzh.ch

We report on a new algorithm for the numeric evaluation of one-loop amplitudes. The method is based on the recursive concatenation of tree structures to build numerator functions as loop-momentum polynomials. Common structures between several Feynman diagrams are identified and reused. The numerator functions are combined with both, tensor integral and OPP reduction to obtain a fast and fully flexible one-loop generator.

*10th International Symposium on Radiative Corrections (Applications of Quantum Field Theory to Phenomenology)
September 26-30, 2011
Mamallapuram, India*

*Speaker.

1. Introduction

The need for precise predictions for many-particle processes at the Large Hadron Collider stimulated a series of recent theoretical developments which led to the completion of various multi-particle next-to-leading-order (NLO) calculations [1–17]. By using tensor-integral reduction and Feynman diagrams, it became possible to handle multi-particle processes with high efficiency and numerical stability [1, 2]. Alternatively, new reductions of on-shell type which avoid tensor integrals and reduce all process-dependent aspects of one-loop calculations to a leading order (LO) problem were introduced [18–20]. While the tensor-reduction approach leads to the fastest numerical codes [1, 2], at present its large-scale applicability is limited by the occurrence of very large algebraic expressions. In contrast, the higher flexibility of the current OPP-based codes [21–23] comes at the price of a lower CPU efficiency.

Here we report on a new one-loop algorithm [24] that naturally adapts to tensor integral and OPP reduction and maximises speed and flexibility in a way that does not depend on the employed reduction. This method is based on the recursive construction of so called *open loops*, which are related to tensor-integral representations of one-loop Feynman diagrams. A recursive algorithm based on tensor integrals was first introduced in the framework of a one-loop Dyson-Schwinger recursion [25].

2. The open loops algorithm

Leading-order transition amplitudes \mathcal{M} and virtual NLO corrections $\delta\mathcal{M}$ are handled as sums of tree and one-loop Feynman diagrams,

$$\mathcal{M} = \sum_d \mathcal{M}^{(d)}, \quad \delta\mathcal{M} = \sum_d \delta\mathcal{M}^{(d)}. \quad (2.1)$$

The corresponding scattering probability densities \mathcal{W} and virtual one-loop corrections $\delta\mathcal{W}$ are

$$\mathcal{W} = \sum_{\text{hel,col}} |\mathcal{M}|^2, \quad \delta\mathcal{W} = \sum_{\text{hel,col}} 2\text{Re}(\mathcal{M}^* \delta\mathcal{M}). \quad (2.2)$$

The sums run over colour and helicity states of each external particle. Colour sums are performed at zero cost by exploiting the *factorisation* of individual diagrams into colour factors $\mathcal{C}^{(d)}$ and colour-stripped amplitudes

$$\mathcal{M}^{(d)} = \mathcal{C}^{(d)} \mathcal{A}^{(d)}, \quad \delta\mathcal{M}^{(d)} = \mathcal{C}^{(d)} \delta\mathcal{A}^{(d)}. \quad (2.3)$$

Four-gluon vertices are split into three contributions for which the factorisation property holds. After algebraic reduction of the colour factors to a standard basis $\{\mathcal{C}_i\}$, all colour sums are encoded in the matrix $\mathcal{K}_{ij} = \sum_{\text{col}} \mathcal{C}_i^* \mathcal{C}_j$, which is computed only once per process (see [26] for details).

Colour-stripped tree diagrams $\mathcal{A}^{(d)}$ are computed by a numerical algorithm that recursively merges sub-trees. We call a sub-tree a subdiagram obtained by cutting a tree. Sub-tree amplitudes are complex n-tuples $w^\beta(i)$, where β is the spinor or Lorentz index of the cut line. The label i represents the topology, momentum and particle content of the sub-tree. Sub-trees are recursively

merged by connecting their cut lines to vertices and propagators:

$$w^\beta(i) = \text{---} \circ \text{---} (i) = \text{---} \circ \text{---} \begin{array}{c} (k) \\ | \\ (j) \end{array} \text{---} . \quad (2.4)$$

The sub-trees i , j and k involve off-shell momenta, but in contrast to off-shell currents they represent individual topologies. Cut lines are marked by dots, and external lines are not depicted. For brevity, quartic vertices are not shown explicitly, but their inclusion is straightforward. In terms of n -tuples, the recursion step reads

$$w^\beta(i) = \frac{X_{\gamma\delta}^\beta(i, j, k) w^\gamma(j) w^\delta(k)}{p_i^2 - m_i^2 + i\epsilon}, \quad (2.5)$$

where $X_{\gamma\delta}^\beta/(p_i^2 - m_i^2 + i\epsilon)$ describes a vertex connecting i , j , k , and a propagator attached to i . The recursion starts with the external lines of a tree, i. e. the wave functions of the scattering particles, and terminates when the sub-trees which are needed to build all tree diagrams have been generated. The algorithm is based on numerical routines that implement all wave functions, propagators and vertices. These building blocks depend only on the theoretical model and are easily obtained from its Feynman rules. This approach is similar to the tree algorithm implemented in MADGRAPH [27]. Its strength lies in the efficiency of colour sums and the systematic *recycling of sub-trees* appearing in different diagrams.

Let us now consider one-loop amplitudes. A colour-stripped n -point loop diagram is an ordered set of n sub-trees, $\mathcal{I}_n = \{i_1, \dots, i_n\}$, connected by loop propagators:

$$\delta\mathcal{A}^{(d)} = \int \frac{d^D q \mathcal{N}(\mathcal{I}_n; q)}{D_0 D_1 \dots D_{n-1}} = \text{---} \begin{array}{c} (i_n) \\ | \\ (i_{n-1}) \\ | \\ (i_1) \\ | \\ (i_2) \end{array} \text{---} . \quad (2.6)$$

The denominators $D_i = (q + p_i)^2 - m_i^2 + i\epsilon$ depend on the loop momentum q , external momenta p_i , and internal masses m_i . All other contributions from loop propagators, vertices, and external sub-trees are summarised in the numerator, which is a polynomial of degree $R \leq n$ in the loop momentum,

$$\mathcal{N}(\mathcal{I}_n; q) = \sum_{r=0}^R \mathcal{N}_{\mu_1 \dots \mu_r}(\mathcal{I}_n) q^{\mu_1} \dots q^{\mu_r}. \quad (2.7)$$

Momentum-shift ambiguities are eliminated by setting $p_0 = 0$. This singles out the D_0 propagator, and the loop momentum q flowing through this propagator is marked by an arrow in (2.6). In traditional one-loop calculations, the coefficients $\mathcal{N}_{\mu_1 \dots \mu_r}$ are explicitly constructed from the Feynman rules, and the amplitude (2.6) is expressed as a linear combination

$$\delta\mathcal{A}^{(d)} = \sum_{r=0}^R \mathcal{N}_{\mu_1 \dots \mu_r}(\mathcal{I}_n) T_{n,r}^{\mu_1 \dots \mu_r} \quad (2.8)$$

The number of coefficients grows with the polynomial degree, which corresponds to the tensorial rank r . However, symmetrising open-loop tensorial indices $\mu_1 \dots \mu_r$ keeps the number of components well under control [25]. Once the coefficients are known, multiple evaluations of the polynomial (2.7) can be performed at a negligible CPU cost [28]. This strongly boosts OPP reduction. Moreover, the same coefficients can be used for a tensor-integral representation of the loop amplitude (2.8). Open loops can thus be interfaced with both OPP and tensor-integral reduction in a natural way.

The efficiency of the open-loop recursion is further increased by means of relations that arise from *pinching loop propagators*. Let us consider the parent (n -point) and child ($(n-1)$ -point) diagrams in Fig. 1, where the child results from pinching the D_{n-1} propagator of the parent. It

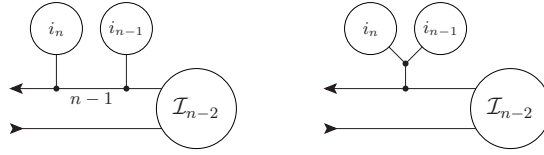


Figure 1: Parent (left) and child (right) open loops which share the common substructure \mathcal{I}_{n-2} .

is evident that the parent can be constructed by *recycling* the \mathcal{I}_{n-2} part of the child. But this requires that parent and child are cut as in Fig. 1. To this end we order the external subtrees using a function $i_k \rightarrow \mathcal{S}(i_k)$ that fulfills $\mathcal{S}(i_k) > 0$; $\mathcal{S}(i_k) \neq \mathcal{S}(i_l)$ if i_k and i_l contain different external legs; $\mathcal{S}(i_k \oplus i_l) > \max\{\mathcal{S}(i_k), \mathcal{S}(i_l)\}$ where $i_k \oplus i_l$ is the merged sub-tree resulting from i_k and i_l . The position and direction of the cut are determined by selecting contiguous sub-trees i_1 and i_n with

$$\mathcal{S}(i_k) > \mathcal{S}(i_1) \quad \forall \quad k > 1, \quad \mathcal{S}(i_n) > \mathcal{S}(i_2). \quad (2.15)$$

This guarantees that parent and child diagrams are cut as in Fig. 1, so that each parent can be constructed from the \mathcal{I}_{n-2} part of a previously computed child.

The possibility of *highly efficient helicity sums* is another key feature of open loops. Unpolarised transition probabilities require multiple evaluations of the polarised amplitudes (2.6). The number of helicity configurations grows exponentially with the particle multiplicity, and the resulting CPU cost can be very large. This can be avoided by exploiting the decomposition (2.8) into helicity-dependent coefficients $\mathcal{N}_{\mu_1 \dots \mu_r}$ and helicity-independent tensor integrals. The CPU expensive evaluation of tensor integrals (2.9) is performed only once, and helicity sums—when restricted to the coefficients—become very fast. More explicitly, the contribution of (2.8) to the unpolarised transition probability is handled as a linear combination

$$\delta\mathcal{W}^{(d)} = \text{Re} \sum_{r=0}^R \delta\mathcal{W}_{\mu_1 \dots \mu_r}^{(d)} T_{n,r}^{\mu_1 \dots \mu_r}, \quad (2.16)$$

with helicity- and colour-summed coefficients

$$\delta\mathcal{W}_{\mu_1 \dots \mu_r}^{(d)} = 2 \sum_{\text{hel}} \left(\sum_{\text{col}} \mathcal{M}^* \mathcal{C}^{(d)} \right) \mathcal{N}_{\mu_1 \dots \mu_r}(\mathcal{I}_n). \quad (2.17)$$

The unpolarised representation (2.16) can be reduced to scalar integrals with any method, including OPP. Within the OPP framework, the reduction is performed by starting from the unpolarised numerator function $\delta\mathcal{W}^{(d)}(\mathcal{I}_n; q) = \sum_r \delta\mathcal{W}_{\mu_1 \dots \mu_r}^{(d)} q^{\mu_1} \dots q^{\mu_r}$; in this way open loops lead to extremely

fast helicity sums as compared to traditional tree generators. The OPP reduction is further improved by combining sets of loop diagrams with identical loop propagators but different external sub-trees.

3. Implementation and Benchmarks

We realised a fully automatic generator of QCD corrections to Standard-Model processes. Diagrams are generated with FEYNARTS [29]; sub-tree and open-loop topologies are processed by a MATHEMATICA program, which concatenates them in a recursive way, reduces colour factors, and returns FORTRAN 90 code. Generalising the setup to other theories than QCD is merely a matter of implementing the corresponding additional vertices and propagators. The reduction to scalar integrals is performed in terms of tensor integrals and, alternatively, with the OPP method. For tensor integrals we use COLLIER, a private library by A. Denner and S. Dittmaier, which implements the scalar integrals of Ref. [30] and reduction methods that avoid instabilities from spurious singularities [31]. The library calculates the coefficients of a covariant decomposition of the tensor integrals and uses them to construct explicit tensor components. OPP reduction is performed with CUTTOOLS [32] and, alternatively, with SAMURAI [33]. Ultraviolet and infrared divergences are dimensionally regularised. While loop denominators are consistently treated in $D = 4 - 2\varepsilon$ dimensions, the momenta q^μ and the coefficients $\mathcal{N}_{\mu_1 \dots \mu_r}$ in (2.7)–(2.9) are handled in $D = 4$. Their $D - 4$ dimensional contributions, which yield so-called R_2 rational terms, are restored via process-independent counterterms [34] using the tree generator.

The correctness of the construction of open loops is verified by a consistency check against our generator for tree amplitudes. By fixing the momentum of the cut propagator in the loop and attaching external wave functions ε_1^α and $\varepsilon_{2\beta}$ to the open loop we obtain pseudo-tree amplitudes

$$\mathcal{P} = \varepsilon_1^\alpha (\mathcal{N}_{\mu_1 \dots \mu_r; \alpha}^\beta q^{\mu_1} \dots q^{\mu_r}) \varepsilon_{2\beta}. \quad (3.1)$$

Agreement between the amplitude \mathcal{P} and the value which is computed independently by evaluating the same diagram with a tree generator confirms the consistent implementation of the routines for the numerical construction and evaluation of open loops as well as the organisation of the recursion and recycling procedures. The loop matrix element is furthermore checked by comparing tensor-integral versus OPP reductions, and observing cancellation of UV and IR divergencies.

To assess flexibility and performance of the method, we considered the $2 \rightarrow 2, 3, 4$ reactions $u\bar{u} \rightarrow W^+W^- + ng$, $u\bar{d} \rightarrow W^+g + ng$, $u\bar{u} \rightarrow t\bar{t} + ng$, and $gg \rightarrow t\bar{t} + ng$, with $n = 0, 1, 2$ gluons. This covers all non-trivial processes of the Les Houches priority list [35]. The open-loop approach leads to compact codes and fast code generation. For instance—as compared to Ref. [2]—the numerical code for $gg \rightarrow W^+W^-b\bar{b}$ becomes two orders of magnitude smaller, and its generation time goes down from more than 1 week to 4 minutes. Also the CPU speed of open loops, when compared against the high performance of Refs. [1, 2], reveals a further improvement. The CPU cost of one-loop scattering probabilities is plotted versus the number of diagrams in Fig. 2 (left). Sums over colours and helicities are always included. For W bosons and top quarks, assuming decays into massless left-handed fermions, we include a single helicity. For the 12 considered processes, involving $\mathcal{O}(10)$ to $\mathcal{O}(10^4)$ diagrams, the CPU cost scales almost linearly with the number of

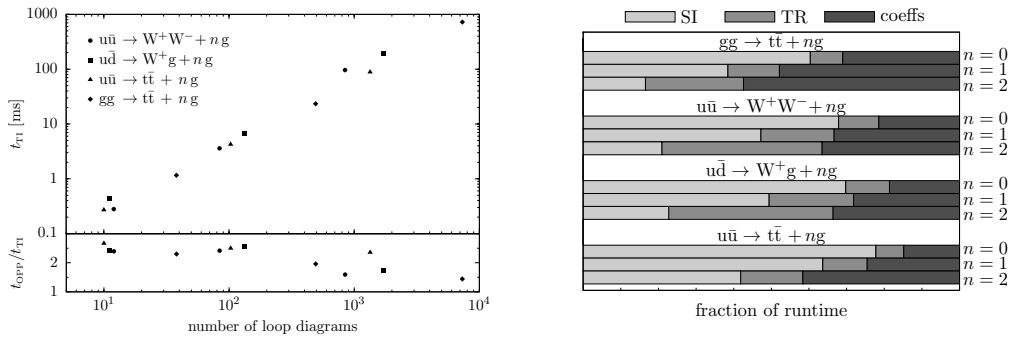


Figure 2: *left:* CPU cost of colour and helicity summed one-loop probabilities $\delta\mathcal{W}$ versus number of diagrams. Runtimes per phase space point, with tensor-integral (t_{TI}) and OPP reduction with CUTTOOLS (t_{OPP}), on a single Intel i5-750 core compiled with ifort 10.1. *right:* Fractions of the CPU time t_{TI} needed for the calculation of scalar integrals (SI), tensor reduction (TR) and for the coefficients $\mathcal{N}_{\mu_1 \dots \mu_r}$ (coeffs).

diagrams. This unexpected feature indicates that the increase of tensorial rank does not represent an additional penalty at large particle multiplicity. With tensor-integral reduction (upper frame), the runtime per phase-space point is typically below 1 ms for $2 \rightarrow 2$ processes; for the most involved $2 \rightarrow 4$ process it never exceeds one second. The ratio of timings obtained with CUTTOOLS and tensor integrals (lower frame) shows that, when combined with open loops, OPP reduction permits to achieve similarly high speed. While always slightly lower, the relative OPP efficiency seems to improve with particle multiplicity. This holds also for SAMURAI.

To study the CPU utilisation in the tensor integral reduction approach in more detail we measured the times needed for the calculation of the scalar integrals, the tensor reduction and the coefficients separately¹. The results are shown in Fig. 2 (*right*). We observe that for low particle multiplicity the time needed for the scalar integrals dominates. For higher multiplicities the relative cost for the reduction of tensor integrals to scalar integrals and for the calculation of the coefficients increases. Typically for a $2 \rightarrow 4$ process the coefficients require around 50% of the CPU time. Still the total runtime is fairly close to the CPU cost of the scalar integrals which represents a lower bound for both methods, tensor integrals and OPP reduction, and is independent of the algorithm employed for the amplitude construction.

4. Summary

We implemented the open loops algorithm and combined it with tensor integral and OPP reduction to obtain a fully flexible generator for one-loop amplitudes. With its excellent CPU speed the method has the potential to handle multi-particle processes with up to $\mathcal{O}(10^5)$ diagrams.

References

- [1] A. Bredenstein, A. Denner, S. Dittmaier, S. Pozzorini, Phys.Rev.Lett. **103** (2009) 012002.

¹Note that there is an ambiguity in splitting the time for the calculation of tensor integrals into individual times for scalar integrals and tensor reduction. We include the reduction of scalar 5- and 6-point functions to lower point functions in the time for the scalar integrals.

- [2] A. Denner, S. Dittmaier, S. Kallweit, S. Pozzorini, *Phys.Rev.Lett.* **106** (2011) 052001.
- [3] G. Bevilacqua et al., *JHEP* **1102** (2011) 083.
- [4] G. Bevilacqua et al., *Phys. Rev. D* **84** (2011) 114017.
- [5] G. Bevilacqua et al, *JHEP* **0909** (2009) 109.
- [6] T. Melia, K. Melnikov, R. Rontsch, G. Zanderighi, *Phys. Rev. D* **83** (2011) 114043.
- [7] R. K. Ellis, K. Melnikov, G. Zanderighi, *JHEP* **0904** (2009) 077.
- [8] N. Greiner, A. Guffanti, T. Reiter and J. Reuter, *Phys. Rev. Lett.* **107** (2011) 102002.
- [9] F. Campanario, C. Englert, M. Rauch, D. Zeppenfeld, *Phys. Lett. B* **704** (2011) 515.
- [10] Z. Bern et al., arXiv:1108.2229 [hep-ph].
- [11] C. F. Berger et al., *Phys. Rev. Lett.* **106** (2011) 092001.
- [12] C. F. Berger et al., *Phys. Rev. D* **82** (2010) 074002.
- [13] C. F. Berger et al., *Phys. Rev. D* **80** (2009) 074036.
- [14] S. Becker et al., *Phys. Rev. Lett.* **108** (2012) 032005.
- [15] Z. Bern et al., arXiv:1112.3940 [hep-ph].
- [16] S. Badger, B. Biedermann, P. Uwer, *Comput. Phys. Commun.* **182** (2011) 1674.
- [17] N. Greiner et al., arXiv:1202.6004 [hep-ph].
- [18] G. Ossola, C. G. Papadopoulos, R. Pittau, *Nucl.Phys.* **B763** (2007) 147-169.
- [19] W. T. Giele, Z. Kunszt, K. Melnikov, *JHEP* **0804** (2008) 049.
- [20] C. Berger et al., *Phys.Rev.* **D78** (2008) 036003.
- [21] A. van Hameren, C. G. Papadopoulos, R. Pittau, *JHEP* **0909** (2009) 106.
- [22] V. Hirschi et al., *JHEP* **1105** (2011) 044.
- [23] G. Cullen et al., arXiv:1111.2034 [hep-ph].
- [24] F. Cascioli, P. Maierhöfer, S. Pozzorini, DOI: 10.1103/PhysRevLett.108.111601.
- [25] A. van Hameren, *JHEP* **0907** (2009) 088.
- [26] A. Bredenstein, A. Denner, S. Dittmaier, S. Pozzorini, *JHEP* **1003** (2010) 021.
- [27] J. Alwall et al., *JHEP* **0709** (2007) 028.
- [28] G. Heinrich, G. Ossola, T. Reiter, F. Tramontano, *JHEP* **1010** (2010) 105.
- [29] T. Hahn, *Comput.Phys.Commun.* **140** (2001) 418-431.
- [30] A. Denner, S. Dittmaier, *Nucl.Phys.* **B844** (2011) 199-242.
- [31] A. Denner, S. Dittmaier, *Nucl.Phys.* **B734** (2006) 62-115.
- [32] G. Ossola, C. G. Papadopoulos, R. Pittau, *JHEP* **0803** (2008) 042.
- [33] P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, *JHEP* **1008** (2010) 080.
- [34] P. Draggiotis, M. V. Garzelli, C. G. Papadopoulos, R. Pittau, *JHEP* **0904** (2009) 072.
- [35] J. R. Andersen et al. (SM and NLO Multileg Working Group), arXiv:1003.1241 [hep-ph].