# Scattering amplitudes at next-to-leading order with Open Loops

**Fabio Cascioli**
*University of Zürich*
*E-mail:* cascioli@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 progress in the implementaion of the open loops method. In the open loops framework the numerator functions of one-loop amplitudes are represented as loop momentum polynomials which are built recursively by concatenating tree-like structures. Combined with tensor integral or OPP reduction this results in a fast and flexible generator for next-to-leading order scattering amplitudes. We study the performance and numerical stability of our implementation and demonstrate numerical agreement with results obtained from independent code.

*"Loops and Legs in Quantum Field Theory" 11th DESY Workshop on Elementary Particle Physics*
*April 15-20, 2012*
*Wernigerode, Germany*

[*]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]. In particular, the OPP method lead to a high degree of automation of one-loop generators [21–23].

Here we report on the open loops algorithm [24], which is based on the recursive numerical construction of numerator functions of one-loop amplitudes as polynomials in the loop momentum. This representation naturally adapts to both, tensor integral and OPP reduction. The algorithm permits us to perform NLO calculations in a highly efficient and numerically stable way by a fully automated program. A recursive algorithm based on tensor integrals was first introduced in the framework of a one-loop Dyson-Schwinger recursion [25].

In section 2 of this contribution we recapitulate the open loops algorithm, followed by a study of the performance and the numerical stability of our implementation in section 3. We perform various internal consistency checks as well as comparisons to independent code. In section 4 we conclude with a summary and a brief outlook.

## 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,

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

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

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

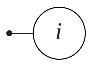
The sums run over colour and helicity states of each external particle. Colour sums are performed at zero cost by exploiting the *factorisation* of individual diagrams into colour factors $\mathcal{C}^{(d)}$ and colour-stripped amplitudes

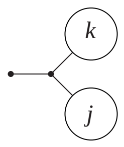
$$\mathcal{M}^{(d)} = \mathcal{C}^{(d)}\mathcal{A}^{(d)}, \qquad \delta\mathcal{M}^{(d)} = \mathcal{C}^{(d)}\delta\mathcal{A}^{(d)}. \tag{2.3}$$

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 $\beta$ 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:

$$w^{\beta}(i) =$$ 

$$=$$ 

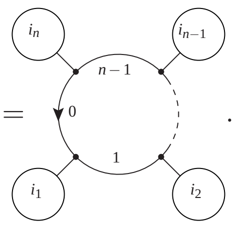
. (2.4)

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

$$w^{\beta}(i) = \frac{X^{\beta}_{\gamma\delta}(i,j,k)\, w^{\gamma}(j)\, w^{\delta}(k)}{p_i^2 - m_i^2 + \mathrm{i}\varepsilon}, \tag{2.5}$$

where $X^{\beta}_{\gamma\delta}/(p_i^2 - m_i^2 + \mathrm{i}\varepsilon)$ 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. 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:

$$\delta\mathcal{A}^{(d)} = \int \frac{\mathrm{d}^D q\, \mathcal{N}(\mathcal{I}_n;q)}{D_0 D_1 \dots D_{n-1}} =$$ 

. (2.6)

The denominators $D_i = (q+p_i)^2 - m_i^2 + \mathrm{i}\varepsilon$ 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,

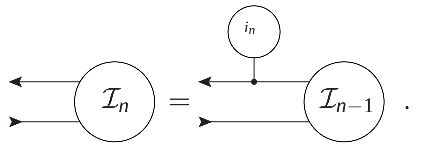
$$\mathcal{N}(\mathcal{I}_n;q) = \sum_{r=0}^{R} \mathcal{N}_{\mu_1\dots\mu_r}(\mathcal{I}_n)\, q^{\mu_1} \dots q^{\mu_r}. \tag{2.7}$$

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 of tensor integrals,

$$\delta\mathcal{A}^{(d)} = \sum_{r=0}^{R} \mathcal{N}_{\mu_1\dots\mu_r}(\mathcal{I}_n)\, T^{\mu_1\dots\mu_r}_{n,r} \quad \text{with} \quad T^{\mu_1\dots\mu_r}_{n,r} = \int \frac{\mathrm{d}^D q\, q^{\mu_1}\dots q^{\mu_r}}{D_0 D_1 \dots D_{n-1}}. \tag{2.8}$$

The tensor integrals $T^{\mu_1\ldots\mu_r}_{n,r}$ are subsequently reduced to $m$-point scalar integrals $T_{m,0}$ with $m = 1,2,3,4$. Alternatively, the OPP method [18] avoids tensor integrals through a direct connection between the numerator $\mathcal{N}(\mathcal{I}_n;q)$ and the scalar-integral representation of the amplitude. The coefficients of the scalar-integrals are determined by evaluating $\mathcal{N}(\mathcal{I}_n;q)$ at loop momenta $q$ that satisfy multiple-cut conditions of the form $D_i = D_j = \cdots = 0$. In this framework, the numerator can be computed with tree-level techniques. Let us consider the *cut loop* that results from (2.6) by cutting the $D_0$ propagator and removing denominators,

$$\mathcal{N}^{\beta}_{\alpha}(\mathcal{I}_n;q) = \quad (2.9)$$



The indices $\alpha$ and $\beta$ are associated with the arrows that mark the ends of the cut line, and the trace of the cut loop corresponds to the numerator $\mathcal{N}(\mathcal{I}_n;q)$. As depicted in (2.9), $n$-point cut loops can be constructed by recursively merging lower-point cut loops and sub-trees. More explicitly,

$$\mathcal{N}^{\beta}_{\alpha}(\mathcal{I}_n;q) = X^{\beta}_{\gamma\delta}(\mathcal{I}_n,i_n,\mathcal{I}_{n-1})\,\mathcal{N}^{\gamma}_{\alpha}(\mathcal{I}_{n-1};q)\,w^{\delta}(i_n), \tag{2.10}$$

where $X^{\beta}_{\gamma\delta}$ and $w^{\delta}$ are the same vertices and sub-trees that enter the tree algorithm. It is thus possible, within the OPP framework, to reduce the calculation of scalar-integral coefficients to a tree-level problem. Highly automatic tree generators can be upgraded to loop generators [21, 22], thereby reducing the human power needed for NLO calculations by orders of magnitude. However, when applied to non-trivial processes, this approach can require massive computing resources. The reason is that OPP reduction requires repeated evaluations of $\mathcal{N}(\mathcal{I}_n;q)$ for a large number of momenta $q$.

This is related to the nature of loop calculations, which requires the knowledge of the numerators as *functions* of the loop momentum $q$. It is thus natural to handle the building blocks of the recursion (2.10) as functions of $q$. Accordingly, the cut loop (2.9) is expressed as a polynomial

$$\mathcal{N}^{\beta}_{\alpha}(\mathcal{I}_n;q) = \sum_{r=0}^{R} \mathcal{N}^{\beta}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_n)\,q^{\mu_1}\ldots q^{\mu_r} \tag{2.11}$$

in the loop momentum $q$. This representation is called an *open loop*. In renormalisable gauge theories, splitting the $X$ tensor in (2.10) into a constant and a linear part, $X^{\beta}_{\gamma\delta} = Y^{\beta}_{\gamma\delta} + q^{\nu}\,Z^{\beta}_{\nu;\gamma\delta}$, leads to recursion relations for $n$-point open loops in terms of lower-point open loops and sub-trees:

$$\mathcal{N}^{\beta}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_n) = \left[Y^{\beta}_{\gamma\delta}\,\mathcal{N}^{\gamma}_{\mu_1\ldots\mu_r;\alpha}(\mathcal{I}_{n-1}) + Z^{\beta}_{\mu_1;\gamma\delta}\,\mathcal{N}^{\gamma}_{\mu_2\ldots\mu_r;\alpha}(\mathcal{I}_{n-1})\right]\,w^{\delta}(i_n). \tag{2.12}$$

Lower point open loops can be reused if they appear in more than one diagram. E. g. when a $(n-1)$-point diagram can be obtained from a $n$-point diagram by pinching one of the loop propagators the diagrams will share a $(n-2)$-point open loop.

The number of coefficients grows with the polynomial degree, which corresponds to the tensorial rank $r$. However, symmetrising open loop tensorial indices $\mu_1\ldots\mu_r$ 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 [27]. 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.

A key feature of open loops is the possibility of *highly efficient helicity sums*. 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 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

$$\delta\mathcal{W}^{(d)} = \mathrm{Re}\sum_{r=0}^{R} \delta\mathcal{W}^{(d)}_{\mu_1\dots\mu_r}\, T^{\mu_1\dots\mu_r}_{n,r}, \tag{2.13}$$

with helicity- and colour-summed coefficients

$$\delta\mathcal{W}^{(d)}_{\mu_1\dots\mu_r} = 2\sum_{\mathrm{hel}}\left(\sum_{\mathrm{col}}\mathcal{M}^*\mathcal{C}^{(d)}\right)\mathcal{N}_{\mu_1\dots\mu_r}(\mathcal{I}_n). \tag{2.14}$$

The unpolarised representation (2.13) 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 $\delta\mathcal{W}^{(d)}(\mathcal{I}_n;q) = \sum_r \delta\mathcal{W}^{(d)}_{\mu_1\dots\mu_r} q^{\mu_1}\dots q^{\mu_r}$; 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 [28]; 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. [29] and reduction methods that avoid instabilities from spurious singularities [30]. 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 [31] and, alternatively, with SAMURAI [32]. Ultraviolet and infrared divergences are dimensionally regularised. While loop denominators are consistently treated in $D = 4-2\varepsilon$ dimensions, the momenta $q^\mu$ and the coefficients $\mathcal{N}_{\mu_1\dots\mu_r}$ in (2.8) are handled in $D = 4$. Their $D-4$ dimensional contributions, which yield so-called $R_2$ rational terms, are restored via process-independent counterterms [33] using the tree generator.

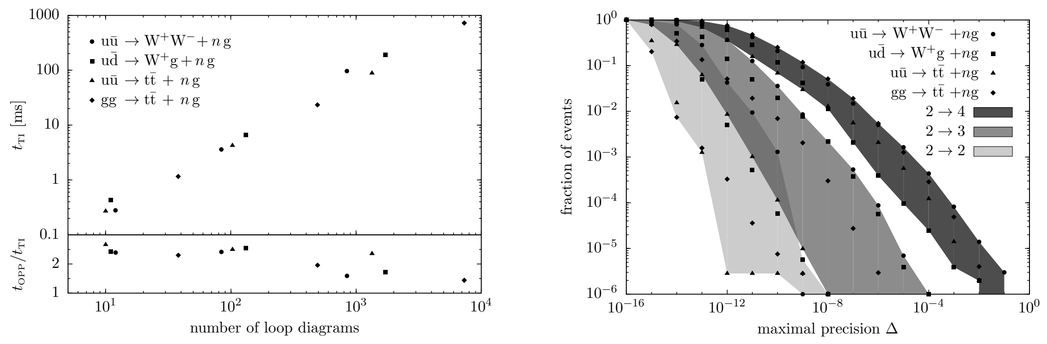


**Figure 1:** *left:* CPU cost of colour and helicity summed one-loop probabilities $\delta\mathcal{W}$ versus number of diagrams. Runtimes per phase space point, with tensor-integral ($t_{\mathrm{TI}}$) and OPP reduction with CUTTOOLS ($t_{\mathrm{OPP}}$), on a single Intel i5-750 core compiled with ifort 10.1. *right:* Accuracy of $\delta\mathcal{W}$ using tensor integral reduction in double precision. The probability of accuracy worse than $\Delta$, in samples of $10^6$ uniformly distributed phase-space points with $\sqrt{s} = 1\,\mathrm{TeV}$, $p_{\mathrm{T}} > 50\,\mathrm{GeV}$, $\Delta R_{ij} > 0.5$, is plotted versus $\Delta$.

To assess the performance and numerical stability of the method, we considered the $2 \to 2,3,4$ reactions $\mathrm{u\bar{u}} \to \mathrm{W^+W^-} + n\mathrm{g}$, $\mathrm{u\bar{d}} \to \mathrm{W^+g} + n\mathrm{g}$, $\mathrm{u\bar{u}} \to \mathrm{t\bar{t}} + n\mathrm{g}$, and $\mathrm{gg} \to \mathrm{t\bar{t}} + n\mathrm{g}$, with $n = 0,1,2$ gluons. This covers all non-trivial processes of the Les Houches priority list [34]. The time to generate and compile the code for a process typically ranges from seconds to a few minutes for processes with up to 6 external particles while the size of the code is at most of the order of 1 MB. In Fig. 1 (*left*) the CPU cost to evaluate one-loop scattering probabilities per phase space point is plotted versus the number of diagrams. Sums over colours and helicities are always included. For W bosons and top quarks we include a single helicity, assuming decays into massless left-handed fermions. 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 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 $2 \to 2$ processes; for the most involved $2 \to 4$ 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. It is instructive to study the relative CPU cost needed for the tensor integrals and the open loops coefficients separately. While the tensor integrals dominate the runtime for simple processes, in complicated cases their contribution reduces to around 50%.

To estimate the numerical accuracy all dimensionful parameters are multiplied by a scale factor $\xi$. This results in scaled scattering probability densities $\delta\mathcal{W}' = \xi^K \delta\mathcal{W}$ where $K$ is the mass dimension of $\delta\mathcal{W}$. After dividing out the scale factor an estimate for the numerical precision is given by the agreement with the original (unscaled) result. Using tensor integral reduction we find an average number of correct digits that ranges from 11 to 15 for the 12 considered processes. For the most involved processes, precision lower than $10^{-5}$ and $10^{-3}$ occurs with less than 2 and 0.1 permille probability, respectively. Fig. 1 (*right*) shows the distribution of the numerical precision for the 12 processes in samples of $10^6$ homogeneously distributed phase space points.

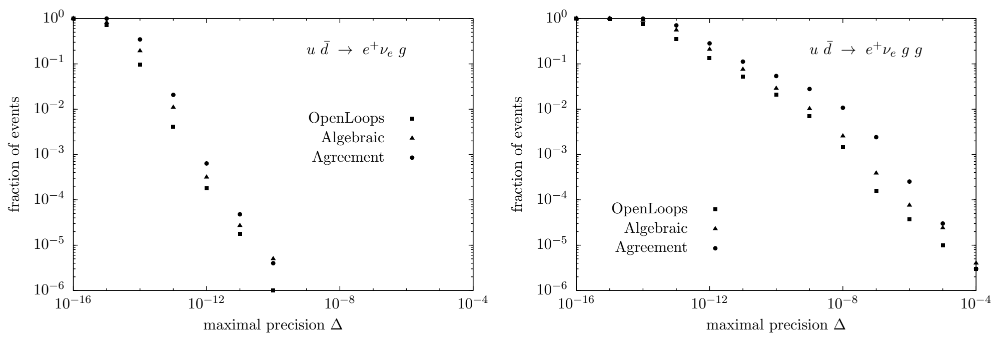


**Figure 2:** Pointwise numerical agreement of open loops and independed algebraic code for the two processes $\mathrm{u}\overline{\mathrm{d}} \to \mathrm{e}^+\overline{\nu}_e\mathrm{g}$ and $\mathrm{u}\overline{\mathrm{d}} \to \mathrm{e}^+\overline{\nu}_e\mathrm{gg}$ in samples of $10^6$ uniformly distributed phase space points. The probability to find numerical agreement worse than $\Delta$ between the two programs is plotted against $\Delta$ along with the accuracy of the individual codes.

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 $\varepsilon_1^\alpha$ and $\varepsilon_{2\beta}$ to the open loop we obtain pseudo-tree amplitudes

$$\mathcal{P} = \varepsilon_1^\alpha \left(\mathcal{N}^\beta_{\mu_1\ldots\mu_r;\alpha}\, q^{\mu_1}\ldots q^{\mu_r}\right)\varepsilon_{2\beta}\,. \tag{3.1}$$

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. Further internal consistency checks include the cancellation of UV and IR divergences, the statisfaction of Ward identities as well as comparing tensor-integral versus OPP reduction.

As an independent check of the entire open loops implementation we compare numerical results for matrix elements obtained with open loops to an in-house generator for one-loop matrix elements which employs algebraic techniques. The high performance of both programs allows us to survey the numerical agreement across $10^6$ phase space points for each process. Fig. 2 visualises the agreement for the two processes $\mathrm{u}\overline{\mathrm{d}} \to \mathrm{e}^+\overline{\nu}_e\mathrm{g}$ and $\mathrm{u}\overline{\mathrm{d}} \to \mathrm{e}^+\overline{\nu}_e\mathrm{gg}$. As of now more than 40 non-trivial processes with four or five external particles were successfully checked.

## 4. Summary and Outlook

The combination of the open loops algorithm with tensor integral and OPP reduction results in 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. Our implementation has proven its reliability by thorough verification of a wide range of processes. As the next step towards an integrated setup for NLO accurate predictions it is necessary to connect the `OpenLoops` program to a Monte Carlo event generator. In order to provide easy access to `OpenLoops` we are working on an interface to `Sherpa` [35] such that the NLO matrix element generation can be controlled via the `Sherpa` user interface. This will open the route to the fully automated generation of NLO predictions for any Standard Model process at the LHC.

## References

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

[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, Phys. Rev. Lett. 108 (2012) 111601.

[25] A. van Hameren, JHEP **0907** (2009) 088.

[26] A. Bredenstein, A. Denner, S. Dittmaier, S. Pozzorini, JHEP **1003** (2010) 021.

[27] G. Heinrich, G. Ossola, T. Reiter, F. Tramontano, JHEP **1010** (2010) 105.

[28] T. Hahn, Comput. Phys. Commun. **140** (2001) 418-431.

[29] A. Denner, S. Dittmaier, Nucl. Phys. **B844** (2011) 199-242.

[30] A. Denner, S. Dittmaier, Nucl. Phys. **B734** (2006) 62-115.

[31] G. Ossola, C. G. Papadopoulos, R. Pittau, JHEP **0803** (2008) 042.

[32] P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, JHEP **1008** (2010) 080.

[33] P. Draggiotis, M. V. Garzelli, C. G. Papadopoulos, R. Pittau, JHEP **0904** (2009) 072.

[34] J. R. Andersen et al. (SM and NLO Multileg Working Group), arXiv:1003.1241 [hep-ph].

[35] T. Gleisberg et al., JHEP **0902** (2009) 007.