Scattering Amplitudes with Open Loops — **Source link**

F. Cascioli, Philipp Maierhöfer, S. Pozzorini

Institutions: University of Zurich

Topics: Scattering amplitude

Related papers:

- Event generation with SHERPA 1.1

- The anti-k_t jet clustering algorithm

- The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations

- A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX

- Parton distributions for the LHC Run II

Year: 2012

Scattering amplitudes with open loops

Cascioli, F ; Maierhöfer, P ; Pozzorini, S

Abstract: We introduce a new technique to generate scattering amplitudes at one loop. Traditional tree algorithms, which handle diagrams with fixed momenta, are promoted to generators of loop-momentum polynomials that we call open loops. Combining open loops with tensor-integral and Ossola-Papadopoulos-Pittau reduction results in a fully flexible, very fast, and numerically stable one-loop generator. As demonstrated with nontrivial applications, the open-loop approach will permit us to obtain precise predictions for a very wide range of collider processes.

Scattering Amplitudes with Open Loops

F. Cascioli,[1] P. Maierhöfer,[1] and S. Pozzorini[1]

[1]*Institut für Theoretische Physik, Universität Zürich, 8057 Zürich, Switzerland*

(Dated: March 6, 2012)

We introduce a new technique to generate scattering amplitudes at one loop. Traditional tree algorithms, which handle diagrams with fixed momenta, are promoted to generators of loop-momentum polynomials that we call open loops. Combining open loops with tensor-integral and OPP reduction results in a fully flexible, very fast, and numerically stable one-loop generator. As demonstrated with non-trivial applications, the open-loop approach will permit to obtain precise predictions for a very wide range of collider processes.

PACS numbers: 11.80.–m, 12.38.Bx, 12.38.Cy

Theoretical simulations of scattering processes play a key role for the interpretation of data collected at the Large Hadron Collider (LHC). Whenever theory predictions are used to link data to model parameters, or to separate signals from backgrounds, perturbative calculations beyond leading order (LO) are indispensable, in order to reduce theoretical errors and to quantify them in a reliable way. The vast physics programme of the LHC requires next-to-leading-order (NLO) predictions for a large variety of processes and theoretical models. In this context, the fairly large particle multiplicities resulting from the high collider energy can lead to one-loop amplitudes of unmanageable complexity. Handling $2 \to 4$ processes with traditional one-loop techniques yields severe numerical instabilities and gigantic algebraic expressions, and can require huge CPU and human power.

The importance of these challenges, marked by the creation of the 2005 Les Houches priority list [1], triggered a series of recent theoretical developments that led to the completion of various multi-particle NLO calculations [2]. By using tensor-integral reduction and Feynman diagrams, it became possible to handle multi-particle processes with high efficiency and numerical stability [3, 4]. Alternatively, new reductions of on-shell type were introduced [5–7] that avoid tensor integrals and reduce all process-dependent aspects of one-loop calculations to a LO problem. In this framework, the Ossola-Papadopoulos-Pittau (OPP) technique [5] led to the development of highly automatic NLO generators [8–10].

One of the features emerging from first LHC applications is a trade-off between CPU efficiency and automation. While the tensor-reduction approach leads to the fastest numerical codes [3, 4], 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 [8–10] comes at the price of a lower CPU efficiency. This motivates us to introduce a new one-loop algorithm 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. Inspired by the observation that colour-ordered multi-gluon amplitudes can be efficiently

computed by combining tensor integrals with a one-loop Dyson-Schwinger recursion [11], we formulate a numerical algorithm that generates one-loop amplitudes via *recursive construction of Feynman diagrams*. As outlined in the following, the method is fully general, and first non-trivial applications demonstrate its high efficiency, when combined with both tensor-integral or OPP reduction.

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

$$\mathcal{M} = \sum_d \mathcal{M}^{(d)}, \qquad \delta\mathcal{M} = \sum_d \delta\mathcal{M}^{(d)}. \tag{1}$$

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

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

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

$$\mathcal{M}^{(d)} = \mathcal{C}^{(d)}\mathcal{A}^{(d)}, \qquad \delta\mathcal{M}^{(d)} = \mathcal{C}^{(d)}\delta\mathcal{A}^{(d)}. \tag{3}$$

Algebraic reduction of the colour factors to a standard basis $\{\mathcal{C}_i\}$ permits to encode all colour sums in the matrix $\mathcal{K}_{ij} = \sum_{\text{col}} \mathcal{C}_i^* \mathcal{C}_j$, which is computed only once per process (see [12] 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:

$$w^\beta(i) = \quad \bullet\!\!-\!\!\boxed{i} \quad = \quad \bullet\!\!-\!\!\!<\begin{smallmatrix} k \\ j \end{smallmatrix} \quad . \tag{4}$$
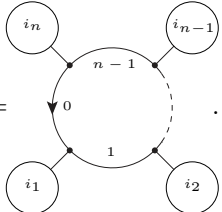
The sub-trees i, j, and k involve off-shell momenta, but in contrast to off-shell currents they represent individual topologies. Cut lines are marked by dots, and external lines are not depicted. For brevity, quartic vertices are not shown explicitly, but their inclusion is straightforward. In terms of n-tuples, the recursion step reads

$$w^\beta(i) = \frac{X^\beta_{\gamma\delta}(i,j,k)\, w^\gamma(j)\, w^\delta(k)}{p_i^2 - m_i^2 + i\varepsilon}, \qquad (5)$$

where $X^\beta_{\gamma\delta}/(p_i^2 - m_i^2 + i\varepsilon)$ 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 generated sub-trees permit to build all tree diagrams. 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 [13]. 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, $\mathcal{I}_n = \{i_1, \ldots, i_n\}$, connected by loop propagators:

$$\delta\mathcal{A}^{(d)} = \int \frac{d^D q\, \mathcal{N}(\mathcal{I}_n; q)}{D_0 D_1 \ldots D_{n-1}} = \quad . \quad (6)$$

The ordering $\{i_1, \ldots, i_n\}$ of the external sub-trees in (6) describes the topology of this particular one-loop Feynman diagram, independently of the coloured or colourless nature of the external particles. Since we do not apply any ordering selection, like e.g. colour ordering, the full set of one-loop diagrams includes all orderings (topologies) that are allowed by the Feynman rules. The denominators $D_i = (q + p_i)^2 - m_i^2 + i\varepsilon$ 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 $R \le n$ in the loop momentum,

$$\mathcal{N}(\mathcal{I}_n; q) = \sum_{r=0}^{R} \mathcal{N}_{\mu_1 \ldots \mu_r}(\mathcal{I}_n)\, q^{\mu_1} \ldots q^{\mu_r}. \qquad (7)$$

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 (6). In traditional one-loop calculations, the coefficients $\mathcal{N}_{\mu_1 \ldots \mu_r}$ are explicitly constructed from

the Feynman rules, and the amplitude (6) is expressed as a linear combination
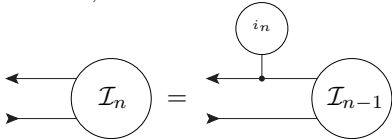
$$\delta\mathcal{A}^{(d)} = \sum_{r=0}^{R} \mathcal{N}_{\mu_1 \ldots \mu_r}(\mathcal{I}_n)\, T_{n,r}^{\mu_1 \ldots \mu_r} \qquad (8)$$

of tensor integrals

$$T_{n,r}^{\mu_1 \ldots \mu_r} = \int \frac{d^D q\, q^{\mu_1} \ldots q^{\mu_r}}{D_0 D_1 \ldots D_{n-1}}. \qquad (9)$$

These latter are subsequently reduced to m-point scalar integrals $T_{m,0}$ with $m = 1, 2, 3, 4$, which originate from (9) by cancelling the numerator and at least $n - 4$ denominators D_i. Alternatively, the OPP method [5] permits to by-pass tensor integrals through a direct connection between the numerator $\mathcal{N}(\mathcal{I}_n; q)$ and the scalar-integral representation of the amplitude. To this end, the numerator is expressed as a polynomial in the denominators D_i. The scalar-integral coefficients are determined by evaluating $\mathcal{N}(\mathcal{I}_n; q)$ at loop momenta q that satisfy multiple-cut conditions of type $D_i = D_j = \ldots = 0$.

In this framework, the numerator can be computed with tree-level techniques. Let us consider the *cut loop* that results from (6) by cutting the D_0 propagator and removing denominators,

$$\mathcal{N}^\beta_\alpha(\mathcal{I}_n; q) = \quad = \quad . \quad (10)$$

The indices α and β 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 (10), n-point cut loops can be constructed by recursively merging lower-point cut loops and sub-trees. More explicitly,

$$\mathcal{N}^\beta_\alpha(\mathcal{I}_n; q) = X^\beta_{\gamma\delta}(\mathcal{I}_n, i_n, \mathcal{I}_{n-1})\, \mathcal{N}^\gamma_\alpha(\mathcal{I}_{n-1}; q)\, w^\delta(i_n), \quad (11)$$

where $X^\beta_{\gamma\delta}$ and w^δ 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 [8, 9], 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 q-momenta.

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 introduce a new kind of loop-generator algorithm, where the building blocks of the recursion (11) are handled as functions of q.

To this end, we express the cut loop (10) as a polynomial

$$\mathcal{N}^\beta_\alpha(\mathcal{I}_n; q) = \sum_{r=0}^{R} \mathcal{N}^\beta_{\mu_1\dots\mu_r;\alpha}(\mathcal{I}_n)\, q^{\mu_1}\dots q^{\mu_r}. \qquad (12)$$

To emphasise the loop-momentum dependence encoded in the set of coefficients $\mathcal{N}^\beta_{\mu_1\dots\mu_r;\alpha}(\mathcal{I}_n)$, we call this representation an *open loop*. In renormalisable gauge theories, splitting the X tensor in (11) into a constant and a linear part,

$$X^\beta_{\gamma\delta} = Y^\beta_{\gamma\delta} + q^\nu\, Z^\beta_{\nu;\gamma\delta}, \qquad (13)$$

we obtain recursion relations for n-point open loops in terms of lower-point open loops and sub-trees:

$$\mathcal{N}^\beta_{\mu_1\dots\mu_r;\alpha}(\mathcal{I}_n) = \Big[Y^\beta_{\gamma\delta}\, \mathcal{N}^\gamma_{\mu_1\dots\mu_r;\alpha}(\mathcal{I}_{n-1})$$
$$+ Z^\beta_{\mu_1;\gamma\delta}\, \mathcal{N}^\gamma_{\mu_2\dots\mu_r;\alpha}(\mathcal{I}_{n-1}) \Big]\, w^\delta(i_n). \qquad (14)$$

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 [11]. Once the coefficients are known, multiple evaluations of the polynomial (7) can be performed at a negligible CPU cost [14]. This strongly boosts OPP reduction. Moreover, the same coefficients can be used for a tensor-integral representation of the loop amplitude (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
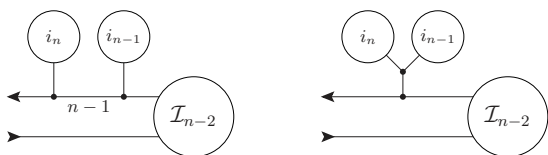
FIG. 1: Parent (left) and child (right) open loops.

child results from pinching the D_{n-1} propagator of the parent. It 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 sub-trees using a function $i_k \to \mathcal{S}(i_k)$ 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

$$\mathcal{S}(i_k) > \mathcal{S}(i_1) \quad \forall \quad k > 1, \qquad \mathcal{S}(i_n) > \mathcal{S}(i_2). \qquad (15)$$

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 (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 (8) into helicity-dependent coefficients $\mathcal{N}_{\mu_1\dots\mu_r}$ and helicity-independent tensor integrals. The CPU expensive evaluation of tensor integrals (9) is performed only once, and helicity sums—when restricted to the coefficients—become very fast. More explicitly, the contribution of (8) to the unpolarised transition probability is handled as a linear combination

$$\delta\mathcal{W}^{(d)} = \mathrm{Re}\left[\sum_{r=0}^{R} \delta\mathcal{W}^{(d)}_{\mu_1\dots\mu_r}\, T^{\mu_1\dots\mu_r}_{n,r} \right], \qquad (16)$$

with helicity- and colour-summed coefficients

$$\delta\mathcal{W}^{(d)}_{\mu_1\dots\mu_r} = 2 \sum_{\mathrm{hel}} \left(\sum_{\mathrm{col}} \mathcal{M}^*\mathcal{C}^{(d)} \right) \mathcal{N}_{\mu_1\dots\mu_r}(\mathcal{I}_n). \qquad (17)$$

The unpolarised representation (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 $\delta\mathcal{W}^{(d)}(\mathcal{I}_n; q) = \sum_r \delta\mathcal{W}^{(d)}_{\mu_1\dots\mu_r} q^{\mu_1}\dots q^{\mu_r}$; 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.

As a proof of concept, we realised a fully automatic generator of QCD corrections to Standard-Model processes. Diagrams are generated with FeynArts [15]; 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. 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. [16] and reduction methods that avoid instabilities from spurious singularities [17]. OPP reduction is performed with Cut-Tools [18] and, alternatively, with Samurai [19]. 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 (7)–(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 [20] using the tree generator.

To assess flexibility and performance of the method, we considered the $2 \to 2, 3, 4$ reactions $\mathrm{u}\bar{\mathrm{u}} \to \mathrm{W}^+\mathrm{W}^- + n\mathrm{g}$,
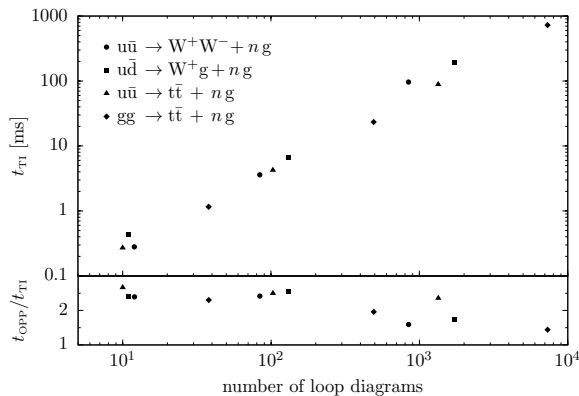
4

FIG. 2: 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 (t_{OPP}), on a single Intel i5-750 core with ifort 10.1.

FIG. 3: Accuracy of $\delta\mathcal{W}$ using tensor reduction in double precision. The probability of accuracy worse than Δ, 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 Δ.

$u\bar{d} \to W^+ g + ng$, $u\bar{u} \to t\bar{t} + ng$, and $gg \to t\bar{t} + ng$, with $n = 0, 1, 2$ gluons. This covers all non-trivial processes of the Les Houches priority list [1]. The open-loop approach leads to compact codes and fast code generation. For instance—as compared to Ref. [4]—the numerical code for $gg \to W^+ W^- b\bar{b}$ 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. [3, 4], reveals a further improvement. The CPU cost of one-loop scattering probabilities is plotted versus the number of diagrams in Fig. 2. 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 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 $1\,\mathrm{ms}$ for $2 \to 2$ processes; for the most involved $2 \to 4$ process it never exceeds one second. The ratio of timings obtained with CUT-TOOLS 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.

The correctness of the results is verified by comparing tensor-integral versus OPP reductions, and checking ultraviolet and infrared cancellations. To assess numerical instabilities, we surveyed the dimensional scaling of probability densities, $\delta\mathcal{W} \to \xi^K \delta\mathcal{W}$, with respect to ξ-variations of mass units. Results obtained with tensor integrals for the 12 considered processes are shown in Fig. 3. In samples of 10^6 phase space points, the av-
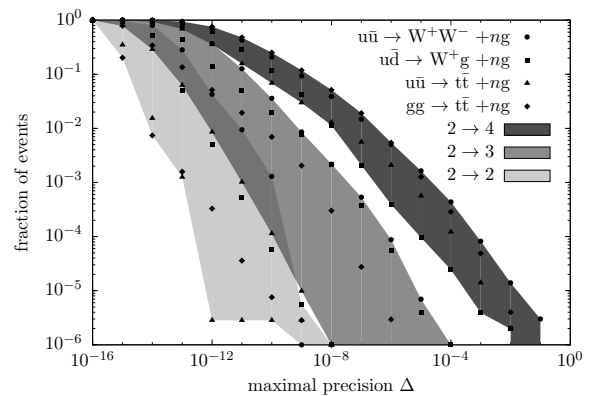
erage number of correct digits for $\delta\mathcal{W}$ ranges from 11 to 15. For the most involved processes, precision lower than 10^{-5} and 10^{-3} occurs with less than 2 and 0.1 permille probability, respectively. This demonstrates the robustness of the tensor-reduction approach [17] in double precision. In contrast, with OPP reduction, a small but non-negligible fraction of points are not sufficiently stable in double precision. A detailed discussion of this aspect, including possible use of quadruple precision or numerical interpolation, is deferred to a forthcoming paper.

In summary, promoting tree generators to open-loop algorithms, we developed a fully flexible, very fast, and numerically stable technique to generate one-loop corrections. Loop momenta are separated from colour and helicity structures in a way that naturally adapts to tensor-integral and OPP reduction, yielding excellent CPU speed with both reductions. Open loops have the potential to address a very wide range of problems at high-energy colliders, ranging from $2 \to 2$ scattering to multi-particle processes with up to $\mathcal{O}(10^5)$ diagrams.

[1] J. R. Andersen et al. (SM and NLO Multileg Working Group) (2010), 1003.1241.
[2] R. Ellis et al. (2011), 1105.4319.
[3] A. Bredenstein et al., Phys.Rev.Lett. **103**, 012002 (2009).
[4] A. Denner et al., Phys.Rev.Lett. **106**, 052001 (2011).
[5] G. Ossola, C. G. Papadopoulos, and R. Pittau, Nucl.Phys. **B763**, 147 (2007).
[6] W. T. Giele, Z. Kunszt, and K. Melnikov, JHEP **0804**, 049 (2008).
[7] C. Berger et al., Phys.Rev. **D78**, 036003 (2008).
[8] A. van Hameren, C. Papadopoulos, and R. Pittau, JHEP

0909, 106 (2009).

[9] V. Hirschi et al., JHEP **1105**, 044 (2011).

[10] G. Cullen et al. (2011), 1111.2034.

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

[12] A. Bredenstein et al., JHEP **1003**, 021 (2010).

[13] J. Alwall et al., JHEP **09**, 028 (2007).

[14] G. Heinrich et al., JHEP **1010**, 105 (2010).

[15] T. Hahn, Comput. Phys. Commun. **140**, 418 (2001).

[16] A. Denner and S. Dittmaier, Nucl. Phys. **B844**, 199

(2011).

[17] A. Denner and S. Dittmaier, Nucl. Phys. **B734**, 62 (2006).

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

[19] P. Mastrolia et al., JHEP **1008**, 080 (2010).

[20] P. Draggiotis et al., JHEP **0904**, 072 (2009).