PROCEEDINGS OF SCIENCE

The next generation of OpenLoops

Philipp Maierhöfer*
Physikalisches Institut, Albert-Ludwigs-Universität Freiburg,
Hermann-Herder-Straße 3, 79104 Freiburg, Germany
E-mail: philipp.maierhoefer@physik.uni-freiburg.de

Mario Prausa
Physikalisches Institut, Albert-Ludwigs-Universität Freiburg,
Hermann-Herder-Straße 3, 79104 Freiburg, Germany
E-mail: mario.prausa@physik.uni-freiburg.de

We report on the development of a new generator for tree and 1-loop matrix elements based on the open-loops algorithm. It is designed to overcome the known shortcomings of the current OpenLoops generator and other existing tools. Key features are a fast on-the-fly process generator which does not require generating process dependent source code, a model agnostic design, supporting the full Standard Model with QCD and electroweak corrections, Beyond-the-Standard Model and Effective Field Theories, as well as interfaces to different reduction libraries, and multi-precision numerics.

Loops and Legs in Quantum Field Theory (LL2018)
29 April 2018 - 04 May 2018
St. Goar, Germany

*Speaker.

PoS(LL2018)057

1. Introduction

The calculation of one-loop amplitudes either for next-to-leading order or loop-induced processes in the Standard Model may now be considered a solved problem from the technical point of view with several publicly available general purpose programs up to the task [1–5]. The tremendous progress in the last decade was set in motion by the transition from algebraic methods to calculate amplitudes to tree-level or tree-like recursive numerical algorithms for loop amplitudes in combination with reduction techniques for the loop integrals or integrands. In particular, the `OpenLoops` method proved to be powerful in terms of performance, both in the generation and evaluation of matrix elements, and numerical stability [6]. It is based on a diagrammatic formulation of the method proposed in [7] for the numerical construction of the numerator of the integrand as loop momentum polynomials, encoding the functional dependence on the loop momentum components. The algorithm has been implemented in the `OpenLoops` program [1] and in `MadLoop` [4]. `Recola` [2] employs a similar approach based on a current recursion instead of Feynman diagrams. For the reduction of one-loop integrals to scalar integrals, several public tools are available, e.g. `Collier` [8] which implements numerically stable tensor integral reduction [9,10], `CutTools` [11] for OPP reduction [12], or `Ninja` [13]. The more recently introduced method of on-the-fly reduction of open-loops [14] which performs integrand reduction already during the construction of the integrand, thereby keeping the tensor rank small, will soon be available in `OpenLoops 2` [15]. It is particularly suited for applications with a demand for high numerical stability like next-to-next-to-leading order real-virtual subtraction.

Now that the technical developments regarding one-loop amplitudes slowed down and automation efforts shifted to two-loop amplitudes [16] it is time to identify and root out shortcomings and inconveniences of the existing tools.

2. A new one-loop generator

2.1 Tree Feynman diagram and amplitude generation

At the time when the work on `OpenLoops` started it was a reasonable choice to rely on `FeynArts` [17] as an existing diagram generator and `Mathematica` to generate process specific source code. However, in the meantime this has become its most apparent weakness. First of all, for state-of-the-art high multiplicity processes, say $2 \to 6$, the generation is displeasingly slow and memory hungry with `FeynArts` requiring 1–2 days and $\sim 30\,\text{GB}$ of memory for a single partonic channel. Another problem which stems from this approach is the necessity to retain a stable interface between the process specific code and the generic code. This hampers the integration of new features that would require interface changes. Therefore we decided to implement our own Feynman diagram generator and abandon the strategy of generating process code, but instead generate processes on-demand in memory. Since this would also require a rewrite of major parts of the process independent code, it lead us to the conclusion that a reimplementation of `OpenLoops` from scratch is the best solution. This new matrix element generator is being developed under the working title `OpenLoops: The Next Generator` (`TNG`). Its diagram generator is based on the recursive construction of tree graphs. To this end, an n-particle tree-level process is regarded as a set of $n-1$ incoming and one outgoing particle. The graphs are generated by recursively

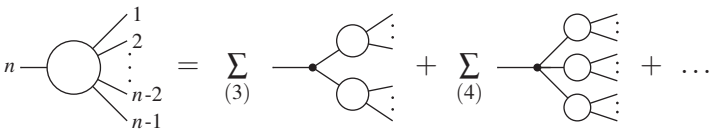


Figure 1: At tree level, Feynman diagrams are generated by recursively partitioning external legs into $2(v-1)$ proper subsets for $3(v)$-point vertices in all possible ways. The recursion stops when the external legs are reached. Then the fields are inserted according to the Feynman rules of the theory.

partitioning the incoming particles according to the vertex degrees in the theory at hand. This is illustrated in figure 1. When the incoming edges are reached, the recursion stops. The tree is then traversed in opposite direction to insert the particle flavours which may appear according to the given Feynman rules. Contributions which lead to the wrong outgoing particle flavour, i.e. not matching the corresponding external particle, are discarded. Furthermore, the coupling orders and colour factors are assigned. Contributions to different coupling orders are kept separate, so that arbitrary interferences may be selected and calculated.

2.2 Colour treatment

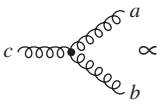
There are several choices for the colour bases which are used in various amplitude generators. The most common ones are

- the trace basis, where each basis element is a product of chains of fundamental generators T^a or traces thereof,
- the colour flow basis with products of Kronecker-Deltas $\delta^{i_a}_{j_a}$, where each adjoint index a is decomposed into a pair of a fundamental and anti-fundamental index, $a = \binom{i_a}{j_a}$,
- no intermediate colour reduction, i.e. the colour factors of the diagrams are kept unreduced until the interference.

Instead of commiting to one particluar choice, the basis may be chosen dynamically depending on the symmetry properties of the amplitude. A linear combination of colour stripped amplitudes A_i, $i = 1,\dots,N$, with (unreduced) colour factors $\mathscr{C}_i$ is expressed as a new linear combination

$$\sum_{i=1}^{N} \mathscr{C}_i A_i = \sum_{i=1}^{n} \mathscr{B}_i \hat{A}_i, \tag{2.1}$$

where $\mathscr{B}_i$, $i = 1,\dots,n$, are the elements of the colour basis, thus defining the colour reduced amplitudes $\hat{A}_i$. The basis may now be chosen in such a way that as many of the $\hat{A}_i$ as possible vanish and therefore do not have to be calculated. The simplest case where this can be achieved is the colour factor f^{abc} of the 3-gluon vertex. Both in the trace basis and the colour flow basis, f^{abc} decomposes into two terms,



$$\propto f^{abc} \propto \underbrace{\mathrm{Tr}(T^a T^b T^c) - \mathrm{Tr}(T^c T^b T^a)}_{\text{trace basis}} \propto \underbrace{\delta^{i_a}_{j_b}\delta^{i_b}_{j_c}\delta^{i_c}_{j_a} - \delta^{i_a}_{j_c}\delta^{i_b}_{j_a}\delta^{i_c}_{j_b}}_{\text{colour flow basis}}. \tag{2.2}$$

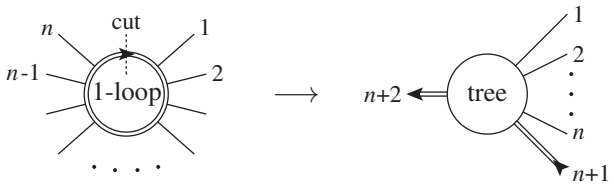


Figure 2: By cutting one of the loop propagators, one-loop diagrams become tree-like with two "special" particles for the loop. n-particle loop amplitudes can therefore be generated similar to $n+2$-particle tree amplitudes. Because of the ambiguity which propagator is cut and due to symmetry properties, there is an overcounting of diagrams which must be removed by appropriate selection rules.

On the other hand, the basis $\{f^{abc}, d^{abc}\}$ has the correct symmetry properties so that the amplitude corresponding to d^{abc} vanishes. In the case of multi-gluon amplitudes this is equivalent to employing the Kleiss–Kuijf relations [18].

2.3 Loop amplitude generation

Feynman diagrams with one loop are generated similarly to tree-level diagrams. By cutting the loop open at one of the loop propagators, the loop diagram becomes tree-like with two special particles corresponding to the cut loop propagator, as illustrated in figure 2. These two particles must be joined later on to close the loop. This strategy leads to a significant amount of redundancy and therefore to an overcounting of diagrams, because for each loop propagator two contributions will be present with this propagator cut and the two possible build directions. This overcounting must be eliminated by an appropriate selection rule, corresponding to the cutting rule of the original open-loops algorithm. At the same time, the cutting rule ensures that structures appearing in more than one diagram are calculated only once and shared among diagrams. Furthermore, certain diagram classes require special treatment. In the common renormalisation schemes, external wave function corrections and tadpole diagrams are discarded. In the case of bubble diagrams with two identical self-adjoint particles in the loop, the symmetry factor $\frac{1}{2}$ must be taken into account.

2.4 Physics model import

A major design element of `TNG` is the possibility to import physics models for Beyond the Standard Model or Effective Field Theory applications. On one hand, this procedure will be completely automated. The models must then be provided in the `UFO` [19] rsp. `NLO-CT` [20] format as generated by `FeynRules` [21]. An extension of `Rept1l` [22] to produce a format readable by `OpenLoops` is planned as well. As of now, the import of tree-level models from `UFO` files has been implemented. On the other hand, the format to specify Feynman rules is human readable and the models can be created or modified at runtime. If a user wants to modify an existing model, e.g. by changing coupling factors, adding or removing specific vertices to or from the theory, this can be achieved without generating and compiling a new model. The only components which must be compiled are the numerical routines corresponding to the spinor and Lorentz structures of the vertices. New numerical routines which are needed by a specific model are generated automatically by the model importer.

	Generate		**Evaluate**		
Process	**TNG**	**OL**	**TNG**	**OL**	**OL/TNG**
pure EW	[ms]	[s]	[μs]	[μs]	
$d\bar{d} \to e^+e^-$	0.075	1	1.47	1.97	1.34
$d\bar{d} \to e^+e^-\gamma$	0.165	1	3.03	3.66	1.21
$d\bar{d} \to e^+e^-\nu_e\bar{\nu}_e$	0.315	2	4.57	5.06	1.11
$d\bar{d} \to e^+e^-\nu_e\bar{\nu}_e\gamma$	0.914	11	9.43	20.6	2.18
$d\bar{d} \to e^+e^-\nu_e\bar{\nu}_e d\bar{d}$	2.94	182	29.1	151	5.19
pure QCD					
$d\bar{d} \to t\bar{t}$	0.081	1	2.24	2.88	1.29
$d\bar{d} \to t\bar{t}g$	0.283	1	4.97	5.19	1.04
$d\bar{d} \to t\bar{t}gg$	1.51	6	30.1	28.8	0.96
$d\bar{d} \to t\bar{t}ggg$	22.9	87	463	568	1.23

Table 1: Timings for the process generation and evaluation of colour and helicity summed tree-level matrix elements on an Intel i7-4790 CPU. `TNG` outperforms `OpenLoops` (OL) in almost all cases, most pronounced in the case of multi-leg electroweak processes. Note that these results are preliminary and subject to future optimisations.

2.5 Technical realisation

Our language of choice to implement `TNG` is `C++`. An important design goal is to produce a maintainable and extensible, well documented code base. On one hand, we use a high level amplitude representation for human readability, on the other hand high performance low level numerics. This is achieved by translating the amplitude representation into a sequence of numerical calls, each one representing an external wave function, a propagator, a vertex or a linear combination of colour stripped amplitudes. This sequence is then executed for each given phase space point rsp. set of parameter values. The calls at this stage are precision independent, so that the same sequence can be used to calculate the matrix element in any supported numerical precision. To this end, we implemented our own memory management. After the amplitude generation, a sufficiently large memory pool which is shared among all processes is allocated to store all wave functions sequentially in memory, avoiding the necessity for (slow) memory allocations during the matrix element evaluation. The numerical precision is in principle not limited to double and quadruple precision, but the data types to be used can be chosen at compile time. However, to the best of our knowledge, there is currently no reduction library available which offers the same flexibility. A similar problem exists with the thread-safety. While `TNG` on its own will be thread-safe, the 1-loop matrix elements will probably only be when we implemented our own reduction machinery, i.e. the on-the-fly reduction of open-loops. For the future we are also planning to explore and utilise the potential of SIMD vectorisation units and GPU computing.

While the tree-level part of the generator is already working, the 1-loop part is still incomplete. Some first benchmarks at tree-level are shown in table 1.

3. Conclusions

A new generator for tree and one-loop matrix elements is in the making: `OpenLoops: The Next Generator`. It is designed to address the shortcomings of `OpenLoops` as well as other available generators. Processes are generated in memory as needed and evaluated in an efficient way in double or higher precision numerics. As known from `OpenLoops`, different libraries may be chosen to perform the reduction to scalar integrals and their evaluation. Our goal is furthermore to provide a future-proof maintainable and extensible framework with a focus on Standard Model and Beyond-the-Standard Model applications. The program will be published as free software when it is complete.

References

[1] F. Cascioli, J. Lindert, P. Maierhöfer and S. Pozzorini, *The openloops one-loop generator*, `https://openloops.hepforge.org`.

[2] S. Actis, A. Denner, L. Hofer, J.-N. Lang, A. Scharf and S. Uccirati, *RECOLA: REcursive Computation of One-Loop Amplitudes*, *Comput. Phys. Commun.* **214** (2017) 140–173, [1605.01090].

[3] A. Denner, J.-N. Lang and S. Uccirati, *Recola2: REcursive Computation of One-Loop Amplitudes 2*, *Comput. Phys. Commun.* **224** (2018) 346–361, [1711.07388].

[4] V. Hirschi, R. Frederix, S. Frixione, M. V. Garzelli, F. Maltoni and R. Pittau, *Automation of one-loop QCD corrections*, *JHEP* **05** (2011) 044, [1103.0621].

[5] G. Bevilacqua, M. Czakon, M. V. Garzelli, A. van Hameren, A. Kardos, C. G. Papadopoulos et al., *HELAC-NLO*, *Comput. Phys. Commun.* **184** (2013) 986–997, [1110.1499].

[6] F. Cascioli, P. Maierhöfer and S. Pozzorini, *Scattering Amplitudes with Open Loops*, *Phys.Rev.Lett.* **108** (2012) 111601, [1111.5206].

[7] A. van Hameren, *Multi-gluon one-loop amplitudes using tensor integrals*, *JHEP* **0907** (2009) 088, [0905.1005].

[8] A. Denner, S. Dittmaier and L. Hofer, *Collier: a fortran-based Complex One-Loop LIbrary in Extended Regularizations*, *Comput. Phys. Commun.* **212** (2017) 220–238, [1604.06792].

[9] A. Denner and S. Dittmaier, *Reduction of one loop tensor five point integrals*, *Nucl. Phys.* **B658** (2003) 175–202, [hep-ph/0212259].

[10] A. Denner and S. Dittmaier, *Reduction schemes for one-loop tensor integrals*, *Nucl. Phys.* **B734** (2006) 62–115, [hep-ph/0509141].

[11] G. Ossola, C. G. Papadopoulos and R. Pittau, *CutTools: A Program implementing the OPP reduction method to compute one-loop amplitudes*, *JHEP* **0803** (2008) 042, [0711.3596].

[12] G. Ossola, C. G. Papadopoulos and R. Pittau, *Reducing full one-loop amplitudes to scalar integrals at the integrand level*, *Nucl.Phys.* **B763** (2007) 147–169, [hep-ph/0609007].

[13] T. Peraro, *Ninja: Automated Integrand Reduction via Laurent Expansion for One-Loop Amplitudes*, *Comput. Phys. Commun.* **185** (2014) 2771–2797, [1403.1229].

[14] F. Buccioni, S. Pozzorini and M. Zoller, *On-the-fly reduction of open loops*, *Eur. Phys. J.* **C78** (2018) 70, [1710.11452].

[15] F. Buccioni, J. Lindert, P. Maierhöfer, S. Pozzorini and M. Zoller, *Openloops 2*, *In preparation*.

[16] *Loops and legs in quantum field theory 2018*, https://indico.desy.de/indico/event/16613/timetable/#all.detailed.

[17] T. Hahn, *Generating Feynman diagrams and amplitudes with FeynArts 3*, *Comput. Phys. Commun.* **140** (2001) 418–431, [hep-ph/0012260].

[18] R. Kleiss and H. Kuijf, *Multi - Gluon Cross-sections and Five Jet Production at Hadron Colliders*, *Nucl. Phys.* **B312** (1989) 616–644.

[19] C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer and T. Reiter, *UFO - The Universal FeynRules Output*, *Comput. Phys. Commun.* **183** (2012) 1201–1214, [1108.2040].

[20] C. Degrande, *Automatic evaluation of UV and R2 terms for beyond the Standard Model Lagrangians: a proof-of-principle*, *Comput. Phys. Commun.* **197** (2015) 239–262, [1406.3030].

[21] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr and B. Fuks, *FeynRules 2.0 - A complete toolbox for tree-level phenomenology*, *Comput. Phys. Commun.* **185** (2014) 2250–2300, [1310.1921].

[22] A. Denner, J.-N. Lang and S. Uccirati, *NLO electroweak corrections in extended Higgs Sectors with RECOLA2*, *JHEP* **07** (2017) 087, [1705.06053].