Computing decay rates for new physics theories with FeynRules and MadGraph5_aMC@NLO

Johan Alwall[a], Claude Duhr[b,*], Benjamin Fuks[c,d], Olivier Mattelaer[e], Deniz Gizem Öztürk[f], Chia-Hsien Shen[a,g]

[a]*Department of Physics, National Taiwan University, Taipei 10617, Taiwan*
[b]*Institute for Particle Physics Phenomenology, University of Durham, Durham, DH1 3LE, United Kingdom*
[c]*Theory Division, Physics Department, CERN, CH-1211 Geneva 23, Switzerland*
[d]*Institut Pluridisciplinaire Hubert Curien/Département Recherches Subatomiques, Université de Strasbourg/CNRS-IN2P3, 23 Rue du Loess, F-67037 Strasbourg, France*
[e]*Center for Cosmology, Particle Physics and Phenomenology, Université Catholique de Louvain, chemin du Cyclotron,2, 1347 Louvain-La-Neuve*
[f]*Institut für Theoretische Physik, Universität Zürich, CH-8057 Zürich, Switzerland*
[g]*California Institute of Technology, Pasadena, CA 91125, USA*

Abstract

We present new features of the FeynRules and MadGraph5_aMC@NLO programs for the automatic computation of decay widths that consistently include channels of arbitrary final-state multiplicity. The implementations are generic enough so that they can be used in the framework of any quantum field theory, possibly including higher-dimensional operators. We extend at the same time the conventions of the Universal FeynRules Output (or UFO) format to include decay tables and information on the total widths. We finally provide a set of representative examples of the usage of the new functions of the different codes in the framework of the Standard Model, the Higgs Effective Field Theory, the Strongly Interacting Light Higgs model and the Minimal Supersymmetric Standard Model and compare the results to available literature and programs for validation purposes.

Keywords: Model building.

[*] Corresponding Author: Claude Duhr; Email: claude.duhr@cern.ch; Phone: +41 (0)22 767 2447

PROGRAM SUMMARY

Manuscript Title: Computing decay rates for new physics theories with FeynRules and MadGraph5_aMC@NLO
Authors: Johan Alwall, Claude Duhr, Benjamin Fuks, Olivier Mattelaer, Deniz Gizem Öztürk, Chia-Hsien Shen
Program Title: MadWidth
Journal Reference:
Catalogue identifier:
Licensing provisions: None.
Programming language: Mathematica & Python.
Computer: Platforms on which Mathematica and Python are available.
Operating system: Operating systems on which Mathematica and Python are available.
Keywords: Model building, Feynman rules, Monte Carlo simulations.
Classification: 11.1 General, High Energy Physics and Computing.
11.6 Phenomenological and Empirical Models and Theories.
External routines/libraries: FeynRules 2.0 or higher.
MadGraph5_aMC@NLO 2.2 or higher.
Nature of problem: The program is a module for the FeynRules and MadGraph5_aMC@NLO packages that allows the computation of tree-level decay widths for arbitrary new physics models. The module consists of two parts:

1. A FeynRules part, which allows one to compute analytically all tree-level two-body decay rates and to output them in the UFO format.
2. A MadGraph5_aMC@NLO part, which allows the numerical computation of many-body decay rates.

Solution method:

1. For the FeynRules part, the analytic expressions for the three-point vertices can be squared to obtain analytic formulas for two-body decay rates.
2. For the MadGraph5_aMC@NLO part, MadGraph is used to generate all Feynman diagrams contributing to the decay, and diagrams that correspond to cascade decays are removed.

Restrictions: Mathematica version 7 or higher. As the package is a module relying on FeynRules and MadGraph5_aMC@NLO all restrictions of these packages apply.
Unusual features: None.
Running time: The computation of the Feynman rules from a Lagrangian, as well as the computation of the decay rates, varies with the complexity of the model, and runs from a few seconds to several minutes. See Section 5 of the present manuscript for more information.

1. Introduction

The Monte Carlo simulation of new physics models with new massive unstable particles requires, to be practically useful, the calculation of the total and partial decay widths for all particles. The higher the number of allowed decay channels, the more daunting it is to do this by hand. Furthermore, depending on the mass hierarchy and interactions among the particles, the computation of two-body decay rates might be insufficient, as higher-multiplicity decays might be the dominant decay modes for some of the particles. Finally, the decay channels that are kinematically allowed are highly dependent on the mass spectrum of the model, so that the decay rates need to be reevaluated for every choice of the input parameters. As a consequence, the computation of all the partial widths of all the particles that appear in a model can be a complex task already at leading order. For this reason, several tools dedicated to the computation of decay rates in the context of specific beyond the Standard Model theories have been developed in the past [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], while many Monte-Carlo event generators are also able to compute partial widths on their own via a dedicated phase-space integration [15, 16, 17, 18, 19, 20, 21, 22, 23]. The aim of this paper is to present a way to compute automatically, using the FeynRules [24, 25] and MadGraph5_aMC@NLO [19, 26] frameworks, all the partial decay widths for large classes of new physics models, in particular when specific width calculators are not publicly available.

FeynRules is a Mathematica package that allows one to compute the interaction vertices of any high-energy physics model directly from its Lagrangian. The package contains a set of translation interfaces to export the Feynman rules to various matrix element generators. The latter often require the widths of all the particles appearing inside the model to be given explicitly and to be defined as numerical input parameters. The workflow to obtain an implementation of a new physics model that can be used for phenomenological studies has so far been the following[1]:

1. obtain an implementation of the model into the matrix element generator of choice (for which an interface to FeynRules exists) where all the widths are set to some default values.
2. for each choice of the numerical input parameters, run the matrix element generator to compute the numerical values of the widths, and insert them back into the model implementation.

While straightforward, it is clear that this process contains a lot of redundant workload. In particular, in many situations the dominant kinematically allowed decay channels are the two-body decays, which can easily be computed by squaring three-point vertices and multiplying by the appropriate phase-space factors. The first main technical advance of this paper is that we present an extension of the FeynRules package that allows one to compute analytically

[1] For the sake of generality, we ignore here external width calculators.

all two-body decay rates, and thus to include their values into the output of the translation interfaces. Moreover, we have extended the Universal FeynRules Output format (UFO) [27], which is the standard interface from FeynRules to MadGraph5_aMC@NLO, GoSam [28] and Herwig++ [17, 29], to include all these analytic formulas so that they can be used dynamically when generating a scattering process.

Two-body decays might however be insufficient and three (or even more)-body decays might be required for a reliable estimation of the total widths of some of the particles. However, computing all the analytic formulae for an arbitrary mass spectrum is beyond reach. In addition, one has to deal with the double counting coming from an intermediate propagator going on-shell, which corresponds to a cascade of two-body decays. Therefore, the second main technical achievement of this paper is to introduce a new MadGraph5_aMC@NLO module, dubbed MadWidth, which determines automatically the required final state multiplicity to reach a given precision on the total width. In addition, it generates the diagrams without any double counting, estimates the contribution of each decay channel and selects those that should be integrated numerically. The inclusion of higher multiplicities may however still be insufficient in cases where loop-induced decay modes are important or in cases where threshold effect are large so that their resummation needs to be taken into account.

The paper is organized as follows: In Section 2 we give a short review of the FeynRules package and we introduce the new functions that can be used to compute two-body decay rates. In Section 3 we present the extension of the UFO format to include the information on these two-body decays. In Section 4 we describe the algorithm implemented in MadWidth to generate all numerically relevant diagrams associated with N-body decay channels and present how the code can be used inside the MadGraph5_aMC@NLO framework. Finally, in Section 5 we validate our implementations by comparing partial widths for four selected models, namely the Standard Model, the Higgs Effective Field Theory [30, 31], the Strongly Interacting Light Higgs model [32] and the Minimal Supersymmetric Standard Model. Our conclusions are presented in Section 6.

2. Automatic computation of two-body decay rates with FeynRules

2.1. The Mathematica *package* FeynRules

In this section we introduce one of the main actors involved in this work, the FeynRules package. We start by presenting a short review of the main functionalities of the package, and we focus on the automatic computation of decay rates within the FeynRules framework in Section 2.2.

In a nutshell, FeynRules is a Mathematica package that allows one for the automatic computation of the Feynman rules of a quantum field theory model directly from its Lagrangian. It can be used with a large variety of physics models involving fields with spin of at most two [33, 34] and/or superfields [35, 36]. The only requirements consist of Lorentz and gauge invariance. In other words, and more technically speaking, all indices appearing inside a

Lagrangian must be correctly contracted. Apart from these restrictions, no further assumptions are made on the functional form of the Lagrangian, so that FeynRules can also be used to compute vertices associated with operators of dimension greater than four and in the context of any gauge choice.

In addition, FeynRules also contains several translation interfaces that allow one to output the Feynman rules into a format readable by various Feynman diagram generators. Currently, dedicated interfaces exist for CalcHep/CompHep [37, 38, 15, 16], FeynArts/FormCalc [39, 40, 41, 42, 43], Sherpa [20, 44] and Whiz-ard/O'Mega [45, 21, 46]. Furthermore, it is also possible to output the model information in the so-called *Universal FeynRules Output* (UFO) format, a format for the implementation of beyond the Standard Model theories into matrix element generators that is not tight to any existing code and that does not make any *a priori* assumption on the structure of the interaction vertices that appear in the model. Finally, FeynRules also comes with specific computational modules that include, *e.g.*, methods dedicated to superspace calculations [35] or mass matrix diagonalization [47].

The implementation of a model into FeynRules does not only require to enter the Lagrangian, but also the definitions of all particles and parameters that appear inside it. In other words, all symbols that are used to write the Lagrangian in Mathematica must be properly declared before Feynman rules may be calculated by FeynRules. The syntax for the declaration of the particles and parameters is based on the original format for model files of FeynArts, extended by new options required by FeynRules. For example, the Standard Model Z-boson can be declared by including in the FeynRules model description,

```
V[1] == {
  ClassName       -> Z,
  SelfConjugate   -> True,
  Mass            -> {MZ, 91.1876},
  Width           -> {WZ, 2.4952}
}
```

This declares a vector field (`V[1]`) denoted by the symbol `Z` that is self-conjugate (*i.e.*, that is equal to its own antiparticle) with a mass M_Z of 91.187 GeV and a width W_Z of 2.4952 GeV. At this stage, the numerical value of the width of the particles must be explicitly given when declaring a new instance of the particle class[2]. The widths are, however, in general not independent input parameters of a model, but related to other parameters of the theory, such as masses and coupling constants. Any consistent phenomenological analysis therefore requires the widths of all the particles appearing in a model to be re-evaluated every time the numerical values of the independent input parameters (consisting of so-called 'benchmark points') are changed. This is at odds with the fact that the widths

[2] If no numerical value is provided through the `Width` attribute of the particle class, a default value of 1 GeV is assigned by FeynRules.

are given as explicit numerical input parameters in a FeynRules model file.

The reason why explicit numerical values need to be specified in a FeynRules model lies in the interfacing to the matrix element generators. Indeed, as the only task of FeynRules is the computation of the tree-level Feynman rules from the Lagrangian, the widths, and even less so their numerical values, are not directly used at any stage by the code. However, when the Feynman rules are exported to one of the aforementioned matrix element generators, these codes often require the widths of the particles to be provided as numerical inputs, thus requiring the corresponding variables to be defined at the FeynRules level and the numerical values computed, *e.g.*, by making use of the generator of choice[3]. Moreover, calculating all the branching ratios of the model particles may also be required by tools such as parton shower programs that are further interfaced to matrix element generators. This has to be repeated for each benchmark point under consideration, which renders the entire approach highly inefficient as it involves a lot of redundant workload.

2.2. Functions dedicated to the computation of decay widths

In this section, we describe the new functionalities of the FeynRules package related to the computation of all two-body decays of a model analytically. We start by giving a very brief review on two-body decays before presenting the new user functions implemented into FeynRules.

The leading-order decay rate of a heavy particle of mass M into N particles of mass m_i is given by

$$\Gamma = \frac{1}{2|M|S} \int \mathrm{d}\Phi_N \, |\mathcal{M}|^2 \ , \tag{1}$$

where S denotes the phase space symmetry factor, $|\mathcal{M}|^2$ the averaged squared matrix element and $\mathrm{d}\Phi_N$ the usual N-body phase-space measure in four dimensions

$$\begin{aligned} \mathrm{d}\Phi_N &= (2\pi)^4 \, \delta^{(4)}\left(P - \sum_{i=1}^N p_i\right) \prod_{i=1}^N \frac{\mathrm{d}^4 p_i}{(2\pi)^3} \, \delta_+(p_i^2 - m_i^2) \\ &= (2\pi)^4 \, \delta^{(4)}\left(P - \sum_{i=1}^N p_i\right) \prod_{i=1}^N \frac{\mathrm{d}^3 p_i}{2\,(2\pi)^3 \, E_i} \ . \end{aligned} \tag{2}$$

In this expression $P = (M, \vec{0})$ stands for the four-momentum of the heavy decaying particle at rest and p_i and E_i are the momenta and energies of the decay products. The absolute value included in Eq. (1), rather unconventional, comes from the fact that in certain beyond the Standard Model theories involving Majorana fermions (*e.g.*, the Minimal Supersymmetric Standard Model), it is

[3] Some tools offer the possibility to compute the widths of all particles on the fly when generating the matrix element associated with a given process. In this case, including the numerical values of the widths at the FeynRules level is not mandatory.

possible to choose the phases of the fermion fields such that the mass is made negative.

In the special case of a two-body decay, $N = 2$, Lorentz invariance implies that the matrix element can only depend on the masses of the external particles, and we can write

$$\Gamma = \frac{\sqrt{\lambda(M^2, m_1^2, m_2^2)}\, |\mathcal{M}|^2}{16\, \pi\, S\, |M|^3}\,, \tag{3}$$

where the Källén function reads $\lambda(M^2, m_1^2, m_2^2) = (M^2 - m_1^2 - m_2^2)^2 - 4m_1^2 m_2^2$. The matrix element of a two-body decay only receives contributions from one single three-point vertex $\mathcal{V}$, and so it can be written as

$$|\mathcal{M}|^2 = \mathcal{V}^{a_1 a_2 a_3}_{\ell_1 \ell_2 \ell_3}\, \mathcal{P}_1^{\ell_1 \ell_1'}\, \mathcal{P}_2^{\ell_2 \ell_2'}\, \mathcal{P}_3^{\ell_3 \ell_3'}\, (\mathcal{V}^*)^{a_1 a_2 a_3}_{\ell_1' \ell_2' \ell_3'}\,, \tag{4}$$

where the color and spin indices of the particle i are denoted by $\ell_i^{(\prime)}$ and a_i. In addition, we have introduced the polarization tensor of the particle i, $\mathcal{P}_i$, which depends on its spin and its mass. As a consequence, the only dependence on the model is through the three-point vertex $\mathcal{V}$ computed by FeynRules, so that we have all the necessary ingredients to evaluate the two-body decay widths analytically. The reason why N-body decays with $N > 2$ are not considered inside the FeynRules framework is due to the fact that, on the one hand, the matrix element does no longer trivially decouple from the phase space and, on the other hand, the remaining phase-space integration might contain infrared divergences for massless particles in the final state. Moreover, the double-counting arising from decay channels of different final-state multiplicities requires a special treatment. The restriction to two-body decays provides in general a good estimate of the width of the particles. In some cases, it is however important to include at least three-body decay contributions. We will discuss how such cases are identified and handled in the context of the MadGraph5_aMC@NLO framework in Section 4.

Using the vertices provided by FeynRules, it is very easy to evaluate Eq. (4) analytically and to obtain the analytic results for *all* two-body decay rates associated with any new physics model. In the rest of this section we describe the new functions included in the FeynRules package that allow one to perform this task on the example of the Standard Model implementation, which is included in the distribution of the package.

In order to compute the partial widths of the particles, it is necessary to first compute the vertices associated with the model in the usual way and to store them in some variable, *e.g.*,

```
vertices = FeynmanRules[ LSM ];
```

where `LSM` denotes the variable containing the Standard Model Lagrangian. Once the vertices have been computed, we can immediately evaluate all two-body decays by issuing the command

```
decays = ComputeWidths[ vertices ];
```

The function ComputeWidths[] first selects all three-point vertices from the list vertices that involve at least one massive particle and no ghost field and/or Goldstone boson. Next, the squared matrix elements are evaluated (in unitary gauge) using Eq. (4) and the results are stored in a list which contains entries of the form

$$\{\{\phi_1,\, \phi_2,\, \phi_3\}\texttt{,}\ \Gamma_{\phi_1\to\phi_2\,\phi_3}\}\,.$$

First, we stress that the output of the function ComputeWidths[] contains the analytic results for the decay rates for *all* possible cyclic rotations of the external particles $\{\phi_1, \phi_2, \phi_3\}$ with a massive initial state, independently whether a given decay channel is kinematically allowed. The reason for this is that, while certain channels might be forbidden/allowed for a chosen set of numerical input parameters, this might not be the case for all possible choices of the external parameters. Second, the output of ComputeWidths[] is also stored internally in the global variable FR$PartialWidth. The use of this global variable will become clear below. Every time the function ComputeDecays[] is executed, the value of the global variable is overwritten, unless the option Save of ComputeWidths[] is set to False (the default is True) and in this case FR$PartialWidth remains unchanged. Summations over possible internal gauge indices in the analytic results are not performed, unless for indices related to the fundamental and adjoint representations of $SU(3)$ in the case where the user employs the conventional symbols for the representation matrices and the corresponding index names presented in the FeynRules manual (see Section 6.1.5 of Ref. [25]).

While all the partial widths for all decay channels (kinematically allowed or not) are stored in the variables decays and FR$PartialWidth, FeynRules contains a set of functions that allows the user to read out directly certain entries of these lists. For example, issuing the command

```
PartialWidth[ {ϕ1,ϕ2,ϕ3 }, decays ]
```

checks, based on the numerical values of the masses defined in the declaration of the particles, whether the decay $\phi_1 \to \phi_2\,\phi_3$ is kinematically allowed, and if so, returns the corresponding partial width $\Gamma_{\phi_1\to\phi_2\,\phi_3}$. The second argument of PartialWidth[] is optional and could be omitted. If omitted, the partial widths stored in the global variable FR$PartialWidth are used by default. Similarly

```
TotWidth[ ϕ1, decays ];
BranchingRatio[ {ϕ1,ϕ2,ϕ3}, decays ];
```

return the analytic results for the total decay rate and the branching ratio, whenever kinematically allowed. Like for PartialWidth[] the second argument is optional. Furthermore, we stress that the analytic expressions can easily be evaluated numerically (using the numerical values for the masses and coupling constants defined in the model) with the built-in FeynRules function NumericalValue[]. Finally, it can be useful to update the information included in the original particle declarations by replacing the numerical value of the widths of all particles by the numerical values obtained with the function TotWidth, which can be achieved by issuing the command

```
UpdateWidths[ decays ];
```

where, as usual, the argument `decays` is optional. After this command has been issued, the updated numerical results for the widths are employed by the translation interfaces to matrix element generators.

3. Extension of the UFO format to include decay information

In the previous section we have seen how FeynRules can be used to compute analytically all tree-level two-body decays associated with a beyond the Standard Model theory, how to update the numerical values for the total widths inside FeynRules and how to export them to matrix element generators. This approach does however not yet solve the problem that widths are not independent parameters, and so they need to be re-evaluated for every benchmark point. While it is in principle possible to rerun FeynRules for each parameter set, this procedure is obviously highly inefficient. In the following, we present an extension of the UFO format [27] that allows one to include the analytic results for the two-body decays. Starting from FeynRules 2.0, the UFO interface automatically includes all two-body decays in the output. It is however possible to disable this feature by setting the option `AddDecays` of the `WriteUFO[]` function to `False` (the default being `True`).

If included, information on the two-body decays is stored into the UFO format in the file `decays.py`. The content of this file contains declarations of instances of the class `Decay` (defined in `object_library.py`). Each instance of this class can be thought of as a collection of analytic formulas for the two-body partial widths of a given particle. For example, the two-body partial widths of the Higgs boson in the Standard Model are represented inside `decays.py` as

```
Decay_H = Decay(name = 'Decay_H',
      particle = P.H,
      partial_widths = {(P.W__minus__,P.W__plus__):'$\Gamma_{H\to W^+W^-}$',
                        (P.Z,P.Z):'$\Gamma_{H\to ZZ}$',
                        (P.b,P.b__tilde__):'$\Gamma_{H\to b\bar{b}}$',
                        (P.ta__minus__,P.ta__plus__):'$\Gamma_{H\to\tau^+\tau^-}$',
                        (P.t,P.t__tilde__):'$\Gamma_{H\to t\bar{t}}$'
      )
```

where $\Gamma_{H\to ij}$ schematically represent the analytic formulas for the partial widths of the Higgs boson. The syntax used to write these analytic formulas in Python form is identical to the syntax used in the UFO format for defining internal parameters, and we refer to Ref. [27] for details. Similar to the output of the `ComputeWidths[]` function described in Section 2.2, *all* possible decays are included, even if kinematically forbidden. In our example[4], this implies that the analytic formula for the decay of a Higgs boson into a top quark pair is also

[4] We consider the first two generations massless.

present (even if not kinematically allowed for a light Higgs boson). It is then up to matrix element generators to filter out at run time the kinematically allowed channels and to combine them consistently into the total width and branching ratios for a given particle.

The example of the Higgs is a case where tree-level two-body decays are not sufficient for an accurate estimation of the width. One must indeed include important contributions arising both from loop-induced diagrams and from three-body decays via off-shell effects. While the inclusion of the loop-induced diagrams lies beyond the scope of this paper, the next section will describe how three-body decays can be handled with MadGraph5_aMC@NLO.

4. MadWidth – a MadGraph5_aMC@NLO module to compute decay widths

MadGraph5_aMC@NLO is a suite of packages related to the computation of matrix elements [19, 48, 49, 50, 51, 52]. Two of its main usages consist of cross section computation and event generation at the leading order (via the MadEvent package [18]) and next-to-leading order (through the MC@NLO framework [26]) in perturbation theory. In addition to their intrinsic accuracy, the precision of the results provided by the aforementioned packages is limited by the precision of the calculations of the numerical values of the model parameters. In particular, the way in which the widths of the particles have been obtained plays a non-negligible role. Although the total widths are in general estimated fairly well if only two-body decays are considered, it is often necessary to include decays to higher-multiplicity final states. In principle, we can generate all N-body tree-level decay processes directly in MadGraph5_aMC-@NLO and find the (partial) widths.[5] However, in addition to the genuine N-body decay processes, this also generates cascade decays and radiation from $(n < N)$-body decays which can be accounted more efficiently from the parent lower-multiplicity decays. Also, often only a small subset of higher-multiplicity final states are relevant for the total width. Both factors hinder a balance between precision and efficiency using MadGraph5_aMC@NLO alone.

MadWidth has been designed with the purpose to solve this issue. It is embedded as a module into MadGraph5_aMC@NLO, and allows the user to compute N-body partial decay widths for arbitrary values of N at tree-level. Assuming the narrow-width approximation[6], the module iteratively builds diagrams from lower multiplicities while at the same time carefully removes contributions from cascade decays and radiative processes. We stress that MadWidth is not the first publicly available code to compute leading-order decay

[5] The same can be done with any Monte-Carlo generator with the same limitation.

[6] Even if those two assumptions are quite generic, there are often particles for which they are not satisfied, such as the Standard Model Higgs boson that has significant loop-induced decay modes or particles whose decay modes are threshold enhanced and where therefore resummation effects are important.

rates, but MadWidth has been designed to go beyond what is done by existing tools like CalcHep, Bridge or Herwig++ in several aspects. First, while CalcHep can evaluate total decay widths on the fly, it only includes contributions from three-body (four-body) decays only if there is no contribution from two-body (three-body) decays. Second, while Herwig++ and Bridge are able to avoid contributions from cascade decays, they are much more limited in the support of beyond the Standard Model theories than MadWidth, which can handle, by its very design, any theory for which a UFO implementation exists.

In the remainder of this section we give a brief account of the algorithms underlying MadWidth. We first explain the algorithm for the diagram generation in Section 4.1, followed by the algorithm to estimate numerical relevance of a given channel compared to a total width and for the selection of the proper set of diagrams for event generation in Sections 4.2 and 4.3. Finally, we introduce how to use the MadWidth module within the MadGraph5_aMC@NLO framework in Section 4.4.

4.1. The MadWidth *algorithm*

The MadWidth module works in an iterative fashion. It begins by generating all two-body decay diagrams, and then iteratively adds extra final state particles to build up higher-multiplicity diagrams[7], until a certain maximal multiplicity $N_{\max}$ is reached. $N_{\max}$ is either provided as an input by the user or determined dynamically such that a requested numerical precision for the total decay width is reached (see Section 4.4 for more details). For N-body final states, all possible diagrams are first generated, including the so-called contact diagrams that contain a single N-point interaction vertex as well as diagrams derived from existing n-body decay processes (with $n < N$). Diagrams corresponding to cascade decay and radiative processes are then removed following a procedure which is detailed in the remainder of this section.

In a first step, the algorithm removes diagrams corresponding to cascade decays. Such diagrams contain at least one intermediate on-shell particle, and they are generated by lower multiplicity processes with one or more subsequent decays. Such diagrams need to be discarded as their contribution to the total width is already accounted for by a lower-multiplicity decay. For example, the decay $t \to bW^+ \to b\ell^+\nu$ is a cascade decay. The internal W-boson is indeed produced on-shell from the $t \to bW^+$ two-body decay, and the W-boson further decays into a neutrino-lepton system. In contrast, diagrams such as the one depicted in Fig. 1 are not defined as cascade decay diagrams and their contribution is thus included into the width calculation. In this case, although the N-body decay diagram contains an n-body decay subdiagram with $n < N$ ($S_1 \to S_2\, S_2$ in the example of the Fig. 1), the N-body final state can only be produced if a internal particle (S_2^* in Fig. 1) is off-shell.

[7] The program identifies stable particles in the very beginning and hence avoids the generation of their decay diagrams. The identification of the stable particles is based on the mass spectrum and the particle interactions.

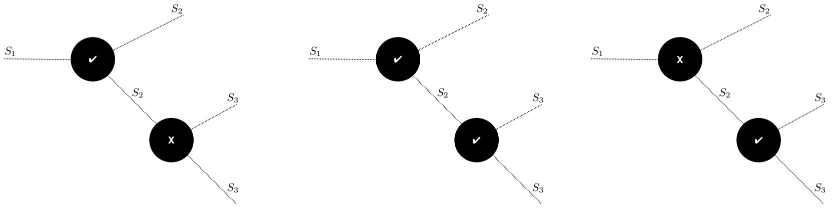

Figure 1: Example of diagrams describing a schematical decay $S_1 \to S_2 S_2^* \to S_2 S_3 S_3$. A cross (tick) on a vertex indicates that the associated decay is kinematically allowed (forbidden) with all three particles being on-shell. In the left-most diagram, the two-body decay $S_1 \to S_2 S_2$ is assumed to be kinematically viable, tagged as 'open', while the process $S_2 \to S_3 S_3$ is kinematically forbidden. Hence, this three-body decay mode of the S_1 particle must be mediated by an off-shell internal S_2^* particle. MadWidth includes this mode as it cannot be captured by a cascade decay of an S_2 particle from a first $S_1 \to S_2 S_2$ decay (see the central diagram), which would have been discarded. In the right-most diagram, $S_1 \to S_2 S_2$ is kinematically forbidden so that S_2 is forced to be off-shell. In this case, the three-body decay is possible and cannot be seen as a cascade of two-body decays, thus it is included by MadWidth.

In order to remove cascade decay diagrams from the width calculation, MadWidth tags all diagrams as 'open' or 'closed' depending on whether or not they are kinematically allowed, and only evaluates N-body diagrams that belong to one of the following categories:

1. The N-body decay diagram contains a closed n-body decay diagram with $n < N$. This configuration is the one of the most common higher-multiplicity decay diagrams.
2. The N-body decay diagram contains an open n-body decay diagram with $n < N$ but the N-body final state can only be produced in the case where one of the internal particles is off-shell (as in Fig. 1).
3. The N-body decay diagram is a contact diagram (made from a single N-point interaction vertex).

Each diagram is then tagged as 'open' or 'closed' and stored for the generation of higher-multiplicity decays. Diagrams which do not belong to any of the above categories are negligible in the narrow width approximation and are omitted.

In a second step, the algorithm discards radiative diagrams. Indeed, as we perform a tree-level (leading order) computation, we do not include any higher-order contributions and an infrared-finite result can only be obtained after coherently discarding all real-emission diagrams. We first investigate the cascade decay topologies and discard any diagram that contains a particle decaying into itself. This forbids, for example, the $W^+ \to W^+ \gamma$, $t \to t\, g$ and $W^+ \to W^+ Z$ transitions. Additionally, we must check that contact decay diagrams (made of a single N-point interaction vertex) do not correspond to a radiative configuration. Equivalently, we need to distinguish N-point interactions that emerge by imposing gauge invariance on a n-point interaction vertex with $n < N$, as it arises when the derivatives of a non-abelian gauge field are replaced by full field strength tensors, from a leading-order new physics contribution. For example the dimension-six operator $H^\dagger\, H\, G_\mu{}^\nu\, G_\nu{}^\mu$ gives rise to both Hgg and $Hggg$ vertices. The second vertex is only necessary to render the first one gauge invariant, and it does therefore not contribute to the width of the Higgs boson into three gluons, because it is a higher-order QCD correction to the $H \to g\,g$ decay. In contrast, if we consider the dimension-eight operator $H^\dagger\, H\, G_\mu{}^\nu\, G_\nu{}^\rho\, G_\rho{}^\mu$, the leading-order $h \to g\,g\,g$ contribution has to be taken into account, even at leading-order in QCD. To disentangle those two cases, we generate all potential diagrams for given initial and final state particles at a specific order in perturbation theory. If a cascade decay topology is found, then the contact decay diagram is considered as radiative and discarded. The definition of the perturbative order of a diagram can be obtained from the UFO library (see Section 6.1.7 of Ref. [25]).

4.2. Fast estimation of N-body partial widths

In order to avoid exporting all the 'open' N-body decay diagrams to MadEvent for numerical integration, MadWidth performs a fast estimation of the contribution of each channel assuming the absence of interference between diagrams contributing to the same final state, and passes to MadEvent only the numerically relevant decay modes[8]. The estimation is based on the formula

$$\Gamma = \frac{1}{2|M|S} \int \mathrm{d}\Phi_N\, |\mathcal{M}|^2 \approx \frac{1}{2|M|S}\, \mathrm{LIPS}_N(M; m_1, \ldots, m_N)\, \langle |\mathcal{M}|^2 \rangle\ , \qquad (5)$$

where $\mathrm{LIPS}_N(M; m_1, \ldots, m_N)$ is the Lorentz-invariant N-body phase-space volume and $\langle |\mathcal{M}|^2 \rangle$ approximates the squared matrix element. In the following we describe in more detail how we evaluate the approximate decay rates.

We start by discussing the evaluation of the phase space volume. Inserting

$$1 = \int \mathrm{d}^4 Q\, \mathrm{d}M_{\mathrm{eff}}^2\, \delta(Q^2 - M_{\mathrm{eff}}^2)\, \delta^{(4)}\left(Q - \sum_{i=1}^{N-1} p_i\right)\ , \qquad (6)$$

into Eq. (2), it is easy to see that (see Fig. 2)

$$\begin{aligned} &\mathrm{LIPS}_N(M; m_1, \ldots, m_N) && (7) \\ &\quad = \int_{M_{\mathrm{min}}^2}^{M_{\mathrm{max}}^2} \frac{\mathrm{M}_{\mathrm{eff}}^2}{2\pi}\, \mathrm{LIPS}_2(M; m_N, M_{\mathrm{eff}})\, \mathrm{LIPS}_{N-1}(M_{\mathrm{eff}}; m_1, \ldots, m_{N-1})\ , \end{aligned}$$

where the two-body phase-space volume is given by

$$\mathrm{LIPS}_2(M; m_1, m_2) = \frac{\sqrt{\lambda(M^2, m_1^2, m_2^2)}}{8\pi M^2}\ . \qquad (8)$$

[8] The interference is only neglected for the estimation of the width. All interference effects are of course correctly taken into account when computing the results for the partial widths.

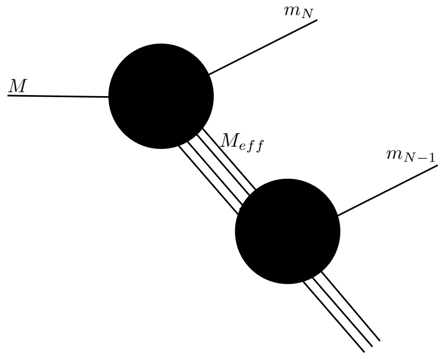

Figure 2: Description of the recursive way of the computation of the phase-space volume factor in Eq. (7). We denote by $M_{\rm eff}$ the effective mass associated with the other $N-1$ particles. See ref. [53, 54] for more details.

Since the residual $(N-1)$-body phase-space measure is Lorentz invariant, we can conveniently calculate it in the rest frame of $P - p_N$. The mass of the effective mother particle (see Fig. 2) is deduced from momentum conservation,

$$M_{\rm eff} = \sqrt{M^2 + m_N^2 - 2M\,E_N}\ , \tag{9}$$

where E_N is the energy of $N^{\rm th}$ particle in the rest frame of the mother particle. This quantity $M_{\rm eff}$ ranges from the production threshold of the remaining N–1 particles ($M_{\rm min} = \sum_{i=1}^{N-1} m_i$) to a maximum value $M_{\rm max} = M - m_N$ corresponding to the case where the $N^{\rm th}$ particle is at rest. Finally, we approximate the exact expression of Eq. (7) by

$$\begin{aligned} {\rm LIPS}_N(M; m_1, \ldots, m_N) \ &\approx c_{ps}(M_{\rm max}^2 - M_{\rm min}^2)\frac{\sqrt{\lambda(M^2, m_N^2, M_{\rm mean}^2)}}{16\pi^2 M^2} \\ &\times {\rm LIPS}_{N-1}(M_{\rm mean}; m_1, \ldots, m_{N-1})\ , \end{aligned} \tag{10}$$

where the average effective mass is defined by $M_{\rm mean} \equiv (M_{\rm max} + M_{\rm min})/2$. Eq. (10) consists of approximating the integral by the area of a rectangle of height $M_{\rm mean}$. As the integrand vanishes at both $M_{\rm max}$ and $M_{\rm min}$ and ${\rm LIPS}_{N-1}(M_{\rm eff}; m_1, \ldots, m_{N-1})$ vanishes at the production threshold, the constant $c_{ps} = 0.8$ has been added to refine the estimation. This formula is recursive and the recursion stops by using the analytic result for the two-body phase space of Eq. (8).

The second ingredient to estimate the decay rate is the average squared matrix element related to the decay process under consideration. This quantity is approximated by mimicking the calculation of standard Feynman diagrams using the dedicated Feynman rules for the propagators and external particles

Table 1: Simplified Feynman rules for propagators and polarization tensors for particles of different spins. We generically denote by E, M and Γ the energy, mass and width of the particle under consideration. Concerning the width, the algorithm either employs an estimation of the two-body decay width, if available, or sets Γ to zero, the subtraction of any resonant contribution allowing one to avoid divergences (see the text for more details). For massive particles, $f(E,M) = 1 + E^2/M^2$, and $f(E,M) = 1$ for massless particles. Our implementation ignores massless propagator as those particles are stable.

Spin	Propagator	Polarization tensor
0	$\frac{1}{(E^2 - M^2 + iM\Gamma)}$	1
$\frac{1}{2}$	$\frac{E}{(E^2 - M^2 + iM\Gamma)}$	$2E$
1	$\frac{1 - \frac{E^2}{M^2}}{(E^2 - M^2 + iM\Gamma)}$	$f(E,M)$
$\frac{3}{2}$	$\frac{\frac{2}{3}(E)(1 - \frac{E^2}{M^2})}{(E^2 - M^2 + iM\Gamma)}$	$2E \cdot f(E,M)$
2	$\frac{\left(\frac{7}{6} - \frac{4}{3}\frac{E^2}{M^2} + \frac{2}{3}\frac{E^4}{M^4}\right)}{(E^2 - M^2 + iM\Gamma)}$	$f(E,M)^2$

given in Table 1. The averaged matrix element square is then given by

$$\langle|\mathcal{M}|^2\rangle = \frac{N_{\rm color}}{N_{\rm s}} \times \left|\prod_{\rm int} {\rm Propa}(E)\right|^2 \times \prod_{\rm ext} \mathcal{P}(E) \times \prod_{\rm vert} \left(\sum_i C_i \, {\rm Lorentz}_i\right)^2 \ . \quad (11)$$

The first factor of this formula includes the average over the initial particle spin states $1/N_{\rm s}$ and the color multiplicity associated with the diagram under consideration $N_{\rm color}$. We recall that only color singlets, triplets, sextets and octets are currently supported by MadGraph5_aMC@NLO.

The second factor in Eq. (11) describes the internal propagators Propa(E) appearing in the diagram, while the third factor includes a product of the polarization tensors $\mathcal{P}(E)$ of the external particles. The Feynman rules of Table 1 have been chosen such as to reproduce the correct results after a summation over all polarization and spin states. Moreover, the available kinetic energy is assumed to be uniformly distributed among the final-state particles, so that the energy E_i of the $i^{\rm th}$ final-state particle of mass M_i reads $E_i = (M - \sum_{j=1}^N M_j)/N + M_i$. The energy associated with an intermediate propagator is then derived from the energy of the particles which it decays into.

Finally, the last factor of Eq. (11) contains the interaction vertices associated with the considered diagram. Each vertex is split up into the different Lorentz structures Lorentz$_j$ that it contains, the corresponding coupling constants being denoted by C_j. The Lorentz structures are further simplified so that they can be evaluated very efficiently. Each object that does not depend on the momenta (Dirac matrices, chirality projectors, *etc.*) is replaced by the identity while each object depending on the momenta (p^μ, $\not{p}$, *etc.*) is replaced by the energy of the relevant particle. We have checked that this treatment is a good approximation also for non-trivial Lorentz structures (such as appearing in new physics models), form factors as well as for derivative couplings.

The decay diagram estimation method has been validated in the framework of various models, and especially in the Standard Model and the Minimal Supersymmetric Standard Model (MSSM). The validation procedure has shown that the estimator can be safely used to select which decay channels are relevant to be computed numerically. Representative examples of the validation tests are presented in Table 2, where we compare the estimation of all partial widths of the heaviest bottom squark $\tilde{b}_2$ to the corresponding exact result derived from a Bridge-MadEvent-CalcHep comparison in the context of the SPS1a MSSM benchmark scenario [55]. One can observe that the estimations reproduce, in all cases, the correct order of magnitude for each channel.

4.3. Estimation of the numerically relevant Feynman diagrams

In Section 4.2, we have presented a method allowing MadWidth to estimate whether a given decay channel is numerically relevant once the corresponding Feynman diagrams have been generated. In this section, we present a second routine dedicated to a rough and quick estimate of the partial width associated with any decay process prior to diagram generation. This allows MadWidth

Table 2: Selection of partial decay widths of the heaviest sbottom in the context of the SPS1a MSSM scenario. We confront the estimations derived by MadWidth to the exact results. We only show partial widths larger than 10^{-7} GeV.

Process	Estimation of MadWidth	Exact result
	[GeV]	[GeV]
$\tilde{b}_2 \to \tilde{\chi}_1^- \, t \, h_1$	1.89e-03	1.04e-03
$\tilde{b}_2 \to \tilde{\chi}_1^- \, t \, Z$	1.21e-03	1.27e-03
$\tilde{b}_2 \to W^- \, t \, \tilde{\chi}_2^0$	9.78e-04	1.36e-03
$\tilde{b}_2 \to \tilde{\chi}_2^- \, b \, W^+$	5.99e-04	1.88e-03
$\tilde{b}_2 \to b \, Z \, \tilde{\chi}_2^0$	1.60e-04	2.88e-04
$\tilde{b}_2 \to \bar{\nu}_\tau \, t \, \tilde{\tau}_2^-$	1.42e-04	3.36e-04
$\tilde{b}_2 \to W^- \, t \, \tilde{\chi}_1^0$	1.34e-04	3.84e-04
$\tilde{b}_2 \to \bar{\nu}_e \, t \, \tilde{e}_L^-$	1.31e-04	3.13e-04
$\tilde{b}_2 \to \bar{\nu}_\mu \, t \, \tilde{\mu}_L^-$	1.31e-04	3.14e-04
$\tilde{b}_2 \to W^- \, b \, \tilde{\chi}_1^+$	1.06e-04	2.04e-04
$\tilde{b}_2 \to \tilde{\nu}_\tau \, t \, \tau^-$	7.52e-05	1.70e-04
$\tilde{b}_2 \to \tilde{\nu}_e \, t \, e^-$	5.21e-05	1.23e-04
$\tilde{b}_2 \to \tilde{\nu}_\mu \, t \, \mu^-$	5.21e-05	1.23e-04
$\tilde{b}_2 \to b \, Z \, \tilde{\chi}_1^0$	2.20e-05	5.39e-05
$\tilde{b}_2 \to b \, h^0 \, \tilde{\chi}_2^0$	1.97e-05	1.33e-05
$\tilde{b}_2 \to W^- \, b \, \tilde{\chi}_2^+$	8.53e-06	5.65e-06
$\tilde{b}_2 \to b \, h^0 \, \tilde{\chi}_1^0$	2.70e-06	2.19e-06
$\tilde{b}_2 \to \bar{b} \, b \, \tilde{b}_1$	1.51e-06	8.58e-07
$\tilde{b}_2 \to \bar{\nu}_\tau \, t \, \tilde{\tau}_1^-$	7.77e-07	2.09e-06
$\tilde{b}_2 \to b \, Z \, \tilde{\chi}_4^0$	6.86e-07	3.54e-07
$\tilde{b}_2 \to \tilde{b}_1^* \, b \, b$	6.37e-07	2.04e-07
$\tilde{b}_2 \to b \, Z \, \tilde{\chi}_3^0$	5.34e-07	3.05e-07
$\tilde{b}_2 \to \bar{u} \, u \, \tilde{b}_1$	2.24e-07	3.48e-07
$\tilde{b}_2 \to \bar{c} \, c \, \tilde{b}_1$	2.24e-07	3.48e-07
$\tilde{b}_2 \to \bar{s} \, s \, \tilde{b}_1$	1.77e-07	4.50e-07
$\tilde{b}_2 \to \bar{d} \, d \, \tilde{b}_1$	1.77e-07	4.50e-07
$\tilde{b}_2 \to \mu^+ \, \mu^- \, \tilde{b}_1$	1.01e-07	1.02e-07
$\tilde{b}_2 \to e^+ \, e^- \, \tilde{b}_1$	1.01e-07	1.02e-07
$\tilde{b}_2 \to \bar{\nu}_e \, \nu_e \, \tilde{b}_1$	5.39e-08	2.03e-07
$\tilde{b}_2 \to \bar{\nu}_\mu \, \nu_\mu \, \tilde{b}_1$	5.39e-08	2.03e-07
$\tilde{b}_2 \to \bar{\nu}_\tau \, \nu_\tau \, \tilde{b}_1$	5.39e-08	2.03e-07

to proceed with diagram generation only in cases where the considered decay mode can yield a non-negligible contribution to the total width.

This new method relies on the derivation of the partial width related to a kinematically allowed $(N+1)$-body decay by combining a N-body decay with the two-body decay of one of the N final-state particles. The average squared matrix element $\langle|\mathcal{M}|^2\rangle_{N+1}$ is hence derived from the knowledge of the average squared matrix elements $\langle|\mathcal{M}|^2\rangle_N$ and $\langle|\mathcal{M}|^2\rangle_i$ that respectively describe the $1 \to N$ decay of the mother particle (of mass M) and the $1 \to 2$ decay of the $i^{\rm th}$ final-state particle (of mass M_i). Calculating $\langle|\mathcal{M}|^2\rangle_N$ as in Eq. (11) after approximately fixing the energy of each final-state particle to $M/(N+1)$, the squared matrix element $\langle|\mathcal{M}|^2\rangle_{N+1}$ can be written as [9]

$$\langle|\mathcal{M}|^2\rangle_{N+1} = \langle|\mathcal{M}|^2\rangle_N \times \langle|\mathcal{M}|^2\rangle_i \times \frac{|\text{Propa}(E_i)|^2}{\mathcal{P}_i(M/(N+1))\mathcal{P}_i(M_i)} \ , \tag{12}$$

where one considers one specific value of i (the summation will be performed below). The last factor of the above formula shows that the external polarization tensors $\mathcal{P}_i^2(M/(N+1))$ and $\mathcal{P}_i^2(M_i)$ of the $i^{\rm th}$ particle that are included in the matrix elements are replaced by a propagator $|\text{Propa}(E_i)|^2$. In addition, the color multiplicity and spin average factor have been neglected.

Since the decay of the $i^{\rm th}$ particle is a two-body decay, the related matrix element is related to the associated partial width as

$$\Gamma_i = \frac{1}{2|M_i|}\text{LIPS}_2(M_i) \times \langle|\mathcal{M}|^2\rangle_i \approx \frac{1}{16\pi|M_i|} \times \langle|\mathcal{M}|^2\rangle_i \ , \tag{13}$$

where $\text{LIPS}_2(M_i) \approx 1/(8\pi)$ is the two-body phase-space volume given in Eq. (8) evaluated by neglecting, for simplicity, all final-state particle masses.

In order to get an estimate of the $N+1$-body decay partial width, it is also necessary to estimate the $N+1$-body phase space volume given by Eq. (10). To this aim, we naively assume that $(M^2_{\rm max} - M^2_{\rm min}) \approx (M/2)^2$ and that $\sqrt{\lambda(M^2, m_N^2, M^2_{\rm mean})} \approx M^2$. Both of these approximations give the correct orders of magnitude and allow one, by iterating, to evaluate the phase space volume as

$$\text{LIPS}_{N+1}(M) \approx c_{ps} \times \left(\frac{M}{8\pi}\right)^2 \text{LIPS}_N(M) \sim \left(c_{ps}\left(\frac{M}{8\pi}\right)^2\right)^{N-2} \times \frac{1}{8\pi} \tag{14}$$

The estimation of the $1 \to N+1$ partial width is then given by combining Eq. (12), Eq. (13), and Eq. (14),

$$\Gamma_{N+1} \approx \widetilde{\Gamma}_N \times \sum_i c_{ps}\Gamma_i \frac{M_i M^2}{4\pi} \frac{|\text{Propa}(E_i)|^2}{\mathcal{P}_i(M_0/(N+1))\mathcal{P}_i(M_i)}, \tag{15}$$

[9] As an order of magnitude estimate, we take $|\text{Propa}(E_i)|^2 \approx 1/(0.5M^2)^2$ for every propagator in $\langle|\mathcal{M}|^2\rangle_N$. This also avoids the propagators to be accidentally on-shell.

where $\widetilde{\Gamma}_N$ is obtained from the phase space volume computed as in Eq. (14) and $\langle|\mathcal{M}|^2\rangle_N$ calculated as described previously. Additionally, the summation over i includes all possible decays of any of the N final-state particles of the $1 \to N$ process.

This method to estimate whether a given process is relevant has been carefully validated in the context of many new physics theories. For illustrative purposes, we focus on the SPS1a MSSM scenario [55] and present in Table 3 the relative contribution of the three-body decay modes to the total width (computed by neglecting any decay to four particles or more) of all massive particles of the model. We compare the exact results to those estimated by MadWidth. This shows that the order of magnitude is either correctly evaluated or overestimated, which is conservative since this implies that the diagrams related to a non-negligible channel are always generated.

4.4. The MadGraph5_aMC@NLO *interface to* MadWidth

The MadWidth module is fully embedded into the MadGraph5_aMC@NLO framework and there are currently two ways to use it. Either MadWidth is directly called from a MadGraph5_aMC@NLO or MadEvent shell, or it is instead run on the fly, at the time of event generation or cross section computation. In all situations, using MadWidth requires a valid UFO model. In the cases where the file `decays.py` is available (see Section 3), the analytic results for two-body decays are directly used by MadWidth. Otherwise, all partial widths are computed numerically.

The first way to use MadWidth is to call it from a MadGraph5_aMC@NLO or MadEvent command interface, respectively initiated via the commands `./bin/mg5_aMC` and `./bin/madevent`. This method is in particular useful for creating a valid `param_card.dat`. Width calculations are performed by issuing the command

```
compute_widths PARTICLE_NAME [OTHER PARTICLE] [OPTIONS]
```

where `PARTICLE_NAME` refers either the name of a particle or its associated Particle Data Group (PDG) code. The user can enter more than one particle name or PDG code, and can also use the keyword `all` to calculate the width of all the particles[10]. The following options are allowed:

- `--body_decay=value` [default: 4.0025]. The code ignores N-body decay contributions when either N is larger than the integer part of `value` or when the estimated error for the total width is lower than the decimal part of `value`. In the case where the integer/decimal part of `value` is set to zero, the associated condition is ignored. For instance,

  > `--body_decay=3` enforces the computation of all two- and three-body decay channels.

[10] It is recommended to compute as many widths as possible in one single execution of the command to reduce the overhead.

Table 3: Relative contribution of the three-body decays to the total width of all massive particles in the context of the SPS1a MSSM scenario. We approximate the total width by neglecting any decay to four particles or more and confront the estimated results of MadWidth (third column) to the exact results (second column). The partial widths of three-body decays are conservatively constrained by our analytical estimation. This shows that the order of magnitude is either correctly evaluated or over-estimated which is conservative.

Particle	$\Gamma_3/(\Gamma_3+\Gamma_2)$	Estimation
t	7.06e-09	4.62e-06
Z	5.16e-07	3.46e-04
W^+	0.00e+00	0.00e+00
h^0	6.76e-02	1.82e+00
H^0	1.55e-03	2.42e-03
A^0	1.30e-03	2.13e-03
H^+	2.53e-03	3.99e-03
$\tilde{d}_L$	8.19e-03	5.92e-03
$\tilde{d}_R$	1.44e-04	9.35e-02
$\tilde{s}_L$	8.15e-03	5.92e-03
$\tilde{s}_R$	1.44e-04	9.35e-02
$\tilde{b}_1$	5.08e-03	1.21e-03
$\tilde{b}_2$	1.07e-02	3.83e-02
$\tilde{u}_L$	9.06e-03	5.46e-03
$\tilde{u}_R$	1.45e-04	2.38e-02
$\tilde{c}_L$	9.08e-03	5.46e-03
$\tilde{c}_R$	1.44e-04	2.38e-02
$\tilde{t}_1$	2.28e-03	1.89e-03
$\tilde{t}_2$	4.74e-03	4.47e-04
$\tilde{e}^-_L$	3.77e-05	2.65e-04
$\tilde{e}^-_R$	1.63e-11	2.08e-06
$\tilde{\mu}^-_L$	3.80e-05	2.65e-04
$\tilde{\mu}^-_R$	1.65e-11	2.08e-06
$\tilde{\tau}^-_1$	0.00e+00	0.00e+00
$\tilde{\tau}^-_2$	5.99e-04	4.14e-04
$\tilde{\nu}_e$	1.26e-08	1.01e-04
$\tilde{\nu}_\mu$	1.27e-08	1.01e-04
$\tilde{\nu}_\tau$	3.76e-04	2.44e-04
$\tilde{g}$	4.94e-04	1.12e-02
$\tilde{\chi}^0_2$	5.08e-03	9.18e-02
$\tilde{\chi}^0_3$	1.16e-03	9.88e-03
$\tilde{\chi}^0_4$	1.34e-03	1.20e-01
$\tilde{\chi}^+_1$	8.09e-03	7.78e-01
$\tilde{\chi}^+_2$	1.45e-03	2.93e-02

`--body_decay=0.01` stops width computations when the estimated error on the total width is lower than 1%.

`--body_decay=3.01` stops a width computation either when all three-body decay contributions have been included or when the estimated error on the total width is lower than 1%.

- `--min_br=value` [default: 0.000625]. If the estimation of the branching ratio associated with a given decay mode is found below `value`, the channel is not integrated numerically and the mode will not appear in the decay table. If not specified explicitly, `value` is set to the decimal part of the `body_decay` parameter divided by four.

- `--precision_channel=value` [default: 0.01]. Required relative precision on each individual channel when integrated numerically.

- `--path=value` [default value for MadGraph5_aMC@NLO: path to the UFO model] [default value for MadEvent: `./Cards/param_card.dat`]. The path to the `param_card.dat` file to use during the numerical evaluation.

- `--output=value` [default: overwrite input file]. The path where to store the new `param_card.dat` file that includes the computed widths.

The second way to use MadWidth is to run it on the fly either through a MadEvent or a aMC@NLO session. Both programs start by checking the `param_card.dat` file. If any of the widths is set to the value `Auto` then MadWidth is called on the fly (with the default options) to evaluate these quantities. The `param_card.dat` file is then overwritten before any further computation. For instance, if the `param_card.dat` file provided to MadEvent contains the lines

```
DECAY   6 Auto # WT
DECAY  23 Auto # WZ
DECAY  24 2.047600e+00 # WW
DECAY  25 5.753088e-03 # WH
```

MadEvent then calls MadWidth to compute the total width and decay table of the top quark and the Z-boson, the widths of the W-boson and Higgs boson remaining unchanged. The original `param_card`.dat file is subsequently overwritten, the above lines being replaced by

```
#      PDG        Width
DECAY  6   1.491472e+00
#  BR             NDA  ID1    ID2   ...
   1.000000e+00   2    24   5 # 1.49147214391
#      PDG        Width
DECAY  23   2.441755e+00
#  BR             NDA  ID1    ID2   ...
```

```
   1.523651e-01   2    3  -3 # 0.372038381506
   1.523651e-01   2    1  -1 # 0.372038381506
   1.507430e-01   2    5  -5 # 0.368077510282
   1.188151e-01   2    4  -4 # 0.290117391009
   1.188151e-01   2    2  -2 # 0.290117391009
   6.793735e-02   2    16  -16 # 0.165886384843
   6.793735e-02   2    14  -14 # 0.165886384843
   6.793735e-02   2    12  -12 # 0.165886384843
   3.438731e-02   2    13  -13 # 0.0839653943458
   3.438731e-02   2    11  -11 # 0.0839653943458
   3.430994e-02   2    15  -15 # 0.0837764784469
#
#      PDG        Width
DECAY  24   2.047600e+00
#
#      PDG        Width
DECAY  25   5.753088e-03
```

where the corresponding partial widths are given under the form of a comment, at the end of each line. Since the code returns not only the total widths, but also all partial widths, the output file is perfectly suitable to be passed to a parton shower program that can then further decay the unstable particles possibly present in the hard events.

5. Illustrative examples

As examples of usage of the tools presented in the previous sections, we focus on two-body and three-body partial width computations and compare, for a set of specific new physics theories, results provided by FeynRules, MadWidth, several public tools and analytic formulas available in the literature.

5.1. Two-body decays

In this section, we focus on two-body decay widths and perform various calculations in the framework of the Standard Model, the Strongly Interacting Light Higgs (SILH) model [32] and the SPS1a MSSM benchmark scenario [55]. We first compute for each of these models all two-body partial widths with FeynRules and then numerically compare the results to those returned by the MadWidth module of MadGraph5_aMC@NLO. Moreover, in the case of the MSSM, we also confront the FeynRules results to the analytic formulas of Ref. [56]. Agreement has been found in all cases, which validates our implementations in particular for theories involving higher dimensional operators (*cf.* the SILH model) and those with Majorana fermions (*cf.* the MSSM). A selection of numerical results can be found in Table 4, Table 5 and Table 6 for the Standard Model, the SILH model and the MSSM, respectively.

Table 4: Selection of partial decay widths in the framework of the Standard Model, as computed by FeynRules and MadWidth.

Decay mode	FeynRules [GeV]	MadWidth [GeV]
$h \to b\,\bar{b}$	0.005390	0.005391
$h \to \tau\,\bar{\tau}$	0.0002587	0.0002587
$h \to c\,\bar{c}$	0.0003967	0.0003967
$W^+ \to e^+\,\nu_e$	0.2225	0.2225
$W^+ \to \tau^+\,\nu_\tau$	0.2223	0.2224
$W^+ \to u\,\bar{d}$	0.6336	0.6336
$W^+ \to c\,\bar{s}$	0.6333	0.6334
$W^+ \to c\,\bar{d}$	0.03401	0.03402
$W^+ \to u\,\bar{s}$	0.03403	0.03403
$Z \to e^-\,e^+$	0.08329	0.08329
$Z \to \tau^-\,\tau^+$	0.0831	0.0831
$Z \to \nu_e\,\bar{\nu}_e$	0.1658	0.1659
$Z \to u\,\bar{u}$	0.2841	0.2842
$Z \to d\,\bar{d}$	0.3667	0.3667
$Z \to c\,\bar{c}$	0.2838	0.2839
$Z \to b\,\bar{b}$	0.3627	0.3628
$t \to b\,W^+$	1.466	1.467

Table 5: Higgs boson partial decay widths in the framework of the SILH model, as computed by FeynRules and MadWidth.

Decay mode	FeynRules [GeV]	MadWidth [GeV]
$h \to \gamma\gamma$	6.447 e-10	6.447 e-10
$h \to g\,g$	7.523 e-06	7.524 e-06
$h \to \gamma\,Z$	4.026 e-11	4.026 e-11

Table 6: Selection of partial decay widths in the framework of the SPS1a MSSM scenario, as computed by FeynRules and MadWidth.

Decay mode	FeynRules [GeV]	MadWidth [GeV]
$\tilde{\chi}_4^0 \to \tilde{\chi}_1^+ W^-$	0.6451	0.6451
$\tilde{\chi}_4^0 \to \tilde{\chi}_1^0 Z$	0.05567	0.05568
$\tilde{\chi}_2^+ \to \tilde{\chi}_1^0 W^+$	0.1682	0.1683
$\tilde{\chi}_2^+ \to \tilde{\chi}_1^+ Z$	0.5755	0.5756
$\tilde{u}_6 \to \tilde{u}_3 Z$	1.39	1.4

5.2. Three-body decays

In this section, we address the calculation of tree-body decay widths and compare results obtained with MadWidth to those available in the literature. We first consider the Higgs Effective Field Theory (HEFT) [30, 31] where the Standard Model Lagrangian is supplemented by an additional dimension-six operator allowing the Higgs boson to directly couple to gluons and photons with coupling strengths tuned to reproduce the corresponding loop-induced vertices of the Standard Model[11]. Next, as a second example, we focus on the MSSM, which allows us to further test the programs in the case of processes with Majorana particles.

In Table 7, we present results for Higgs boson decays in the framework of the HEFT model and compare partial widths calculated by MadWidth to results returned by SMCalc[12]. For the sake of completeness, we have included both two-body and three-body decay channels and found good agreement between the two programs.

In Table 8, we consider the SPS1a MSSM scenario and compare the results obtained with the `decay.py` file generated by FeynRules (that only includes two-body decay modes), with MadWidth when using all the default option values described in Section 4.4, with MadWidth by enforcing the calculation of all three-body decay channels and with Bridge (when including all three-body decay modes). Agreement below the percent level has been found. Additionally, this shows that for the SPS1a MSSM scenario, three-body decay contributions to the total widths are negligible in a very good approximation.

Finally, we compare the speed of MadWidth (using the `decay.py` file gen-

[11] We stress that only these dimension-six operators are included, and not the full set of such operators.

[12] SMCalc is a program that contains analytical formulæ for all leading-order SM partial widths. It is available from Ref. [8].

Table 7: Higgs boson partial decay widths in the framework of the HEFT model, as computed by MadWidth and by SMCalc.

Decay mode:	MadWidth [GeV]	SMCalc [GeV]
$h \to b\,\bar{b}$	0.00430	0.00430
$h \to c\bar{c}$	0.000496	0.000496
$h \to \tau\,\bar{\tau}$	0.000259	0.000259
$h \to g\,g$	0.000195	0.000195
$h \to W\,W^* \to W\,f\,f$	0.000771	0.000775
$h \to Z\,Z^* \to Z\,f\,f$	8.44e-05	8.40e-05
$h \to \gamma\,\gamma$	9.70e-06	9.731e-06

Table 8: Selection of total decay widths in the framework of the SPS1a MSSM scenario, as computed from the `decay.py` file generated by FeynRules (first column), MadWidth with default option values (second column) and by enforcing three-body decays (third column) and by Bridge (fourth column).

Particle	FeynRules	MadWidth	MadWidth	Bridge
	Two-body [GeV]	Default [GeV]	Three-body [GeV]	Three-body [GeV]
$\tilde{\chi}_1^+$	1.704e-02	1.718e-02	1.718e-02	1.724e-02
$\tilde{\chi}_2^+$	2.487	2.488	2.488	2.485
H^+	6.788e-01	6.788e-01	6.802e-01	6.780e-01
$\tilde{b}_1$	3.736	3.736	3.740	3.731
$\tilde{b}_2$	8.016e-01	8.071e-01	8.094e-01	8.100e-01
$\tilde{\chi}_2^0$	2.078e-02	2.087e-02	2.088e-02	2.082e-02
$\tilde{\chi}_3^0$	1.916e+00	1.916e+00	1.916e+00	1.914e+00

Table 9: Time necessary to compute all particle total widths in the context of the HEFT model and the SPS1a MSSM scenario by making use of the `decay.py` file generated by FeynRules only (first column), MadWidth with all default option values and Bridge by restricting the computation to two-body decays only (second column) and after including three-body decays too (fourth column). For the tests, we have employed a machine with a 2.3 GHz Intel Core i7 processor and 16 GB of memory (1600 MHz DDR3).

Model	FeynRules	Bridge	MadWidth	Bridge
	Two-body	Two-body	Default	Three-body
HEFT model	0.6 s	60s	40s	114 s
SPS1a MSSM scenario	12 s	13min 43s	84 s	1h47

erated by FeynRules) to the one of Bridge and indicate in Table 9 the time necessary to compute the total widths of all model particles in the context of both the HEFT model and the SPS1a MSSM scenario. When only the analytic formulas for the two-body decays (implemented in the `decay.py` file generated by FeynRules) are used (first column of the table), MadWidth turns out to be much faster than Bridge (second column of the table) as no diagram generation has been required at all for the first case. The time necessary to generate the `decay.py` UFO file has however not been included (a few seconds and minutes for the HEFT model and MSSM, respectively) as this has to be performed only once for each model. When including three-body decay contributions, one can note the formidable gain in time when using MadWidth instead of Bridge (last two columns of the table) thanks to the usage of the fast estimator of MadWidth which allows one to only compute numerically relevant diagrams.

6. Conclusion

In this paper, we have presented new routines of the FeynRules package dedicated to the computation of two-body partial widths, so that the latter can now be calculated automatically and analytically from the knowledge of the Lagrangian alone. The UFO format has been extended accordingly to include all the relevant information. This extension is currently supported by MadGraph5_aMC@NLO, which uses it to calculate particle widths at run-time. In addition, a new module of MadGraph5_aMC@NLO, named MadWidth, has been developed with the aim of computing N-body decay widths in full generality and in an efficient manner (possibly at run-time when using the matrix-element generator). The MadWidth routines automatically remove all the subprocesses that are numerically negligible, tune the final-state multiplicity to reach a given target precision on the total widths and avoid the double-counting of any channel. All the computations are done at tree level/leading order and rely on the narrow width approximation. As such, MadWidth cannot be used to obtain reliable predictions for the widths in cases where higher-order effect are important. Our codes have been carefully validated against existing older programs and published results in the literature in the case of the Standard Model, Higgs Effective Field Theories, the Strongly Interacting Light Higgs model and the Minimal Supersymmetric Standard Model. The widths obtained by MadWidth are accurate enough to be used in any LO Monte-Carlo generator as long as the narrow width approximation holds.

Acknowledgments

We want to thank C. Degrande and F. Maltoni for useful discussions on this project and D. Goncalves Netto and K. Mawatari for their constructive bug reports. O.M. want to thank the IPPP center for its hospitality during the time of this project. B.F. was supported in part by the French ANR 12 JS05 002 01 BATS@LHC and by the Theory-LHC-France initiative of the CNRS/IN2P3. CHS is supported by the DOE under grant number de-sc0010255. DGO is supported by Theoretische Forschungen auf dem Gebiet der Elementarteilchen (SNF). OM is 'Chercheur scientifique logistique postdoctoral F.R.S.-FNRS', Belgium. This work was partly supported by the Research Executive Agency (REA) of the European Union under the Grant Agreement number PITN-GA-2010-264564 (LHCPhenoNet), by MCnetITN FP7 Marie Curie Initial Training Network PITN-GA-2012-315877, by the IISN "MadGraph" convention 4.4511.10, by the IISN "Fundamental interactions" convention 4.4517.08, and in part by the Belgian Federal Science Policy Office through the Interuniversity Attraction Pole P7/37.

References

[1] A. Djouadi, J. Kalinowski, M. Spira, HDECAY: A Program for Higgs boson decays in the standard model and its supersymmetric extension, Comput.Phys.Commun. 108 (1998) 56–74. arXiv:hep-ph/9704448, doi:10.1016/S0010-4655(97)00123-9.

[2] S. Heinemeyer, W. Hollik, G. Weiglein, FeynHiggs: A Program for the calculation of the masses of the neutral CP even Higgs bosons in the MSSM,

Comput. Phys. Commun. 124 (2000) 76–89. arXiv:hep-ph/9812320, doi:10.1016/S0010-4655(99)00364-1.

[3] B. C. Allanach, SOFTSUSY: a program for calculating supersymmetric spectra, Comput. Phys. Commun. 143 (2002) 305–331. arXiv:hep-ph/0104145, doi:10.1016/S0010-4655(01)00460-X.

[4] M. Muhlleitner, A. Djouadi, Y. Mambrini, SDECAY: A Fortran code for the decays of the supersymmetric particles in the MSSM, Comput.Phys.Commun. 168 (2005) 46–70. arXiv:hep-ph/0311167, doi:10.1016/j.cpc.2005.01.012.

[5] W. Porod, SPheno, a program for calculating supersymmetric spectra, SUSY particle decays and SUSY particle production at e+ e- colliders, Comput. Phys. Commun. 153 (2003) 275–315. arXiv:hep-ph/0301101, doi:10.1016/S0010-4655(03)00222-4.

[6] A. Djouadi, M. Muhlleitner, M. Spira, Decays of supersymmetric particles: The Program SUSY-HIT (SUspect-SdecaY-Hdecay-InTerface), Acta Phys.Polon. B38 (2007) 635–644. arXiv:hep-ph/0609292.

[7] P. Meade, M. Reece, BRIDGE: Branching ratio inquiry / decay generated eventsarXiv:hep-ph/0703031.

[8] http://madgraph.hep.uiuc.edu/calculators.html.

[9] D. Eriksson, J. Rathsman, O. Stal, 2HDMC: Two-Higgs-Doublet Model Calculator Physics and Manual, Comput. Phys. Commun. 181 (2010) 189–205. arXiv:0902.0851, doi:10.1016/j.cpc.2009.09.011.

[10] W. Frisch, H. Eberl, H. Hlucha, HFOLD - a program package for calculating two-body MSSM Higgs decays at full one-loop level, Comput. Phys. Commun. 182 (2011) 2219–2226. arXiv:1012.5025, doi:10.1016/j.cpc.2011.05.007.

[11] D. Das, U. Ellwanger, A. M. Teixeira, NMSDECAY: A Fortran Code for Supersymmetric Particle Decays in the Next-to-Minimal Supersymmetric Standard Model, Comput.Phys.Commun. 183 (2012) 774–779. arXiv:1106.5633, doi:10.1016/j.cpc.2011.11.021.

[12] H. Hlucha, H. Eberl, W. Frisch, SFOLD - a program package for calculating two-body sfermion decays at full one-loop level in the MSSM, Comput. Phys. Commun. 183 (2012) 2307–2312. arXiv:1104.2151, doi:10.1016/j.cpc.2012.05.022.

[13] R. Contino, M. Ghezzi, C. Grojean, M. Muhlleitner, M. Spira, Effective Lagrangian for a light Higgs-like scalar, JHEP 1307 (2013) 035. arXiv:1303.3876, doi:10.1007/JHEP07(2013)035.

[14] J. Baglio, R. Gröber, M. Mühlleitner, D. T. Nhung, H. Rzehak, M. Spira, J. Streicher, K. Walz, NMSSMCALC: A Program Package for the Calculation of Loop-Corrected Higgs Boson Masses and Decay Widths in the (Complex) NMSSM, Comput. Phys. Commun. 185 (12) (2014) 3372–3391. `arXiv:1312.4788`, `doi:10.1016/j.cpc.2014.08.005`.

[15] A. Pukhov, CalcHEP 3.2: MSSM, structure functions, event generation, batchs, and generation of matrix elements for other packages, `arXiv:hep-ph/0412191`.

[16] A. Belyaev, N. D. Christensen, A. Pukhov, CalcHEP 3.4 for collider physics within and beyond the Standard Model, Comput.Phys.Commun. 184 (2013) 1729–1769. `arXiv:1207.6082`, `doi:10.1016/j.cpc.2013.01.014`.

[17] M. Bahr, S. Gieseke, M. Gigg, D. Grellscheid, K. Hamilton, et al., Herwig++ Physics and Manual, Eur.Phys.J. C58 (2008) 639–707. `arXiv:0803.0883`, `doi:10.1140/epjc/s10052-008-0798-9`.

[18] F. Maltoni, T. Stelzer, MadEvent: Automatic event generation with MadGraph, JHEP 0302 (2003) 027. `arXiv:hep-ph/0208156`.

[19] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, T. Stelzer, MadGraph 5 : Going Beyond, JHEP 1106 (2011) 128. `arXiv:1106.0522`, `doi:10.1007/JHEP06(2011)128`.

[20] T. Gleisberg, et al., SHERPA 1.alpha, a proof-of-concept version, JHEP 02 (2004) 056. `arXiv:hep-ph/0311263`.

[21] W. Kilian, T. Ohl, J. Reuter, WHIZARD: Simulating Multi-Particle Processes at LHC and ILC, Eur. Phys. J. C71 (2011) 1742. `arXiv:0708.4233`, `doi:10.1140/epjc/s10052-011-1742-y`.

[22] T. Hahn, J. I. Illana, Excursions into FeynArts and FormCalc, Nucl. Phys. Proc. Suppl. 160 (2006) 101–105. `arXiv:hep-ph/0607049`, `doi:10.1016/j.nuclphysbps.2006.09.035`.

[23] M. Klasen, Calculating two and three-body decays with FeynArts and FormCalc, Int.J.Mod.Phys. C14 (2003) 1273–1278. `arXiv:hep-ph/0210426`, `doi:10.1142/S012918310300539X`.

[24] N. D. Christensen, C. Duhr, FeynRules - Feynman rules made easy, Comput. Phys. Commun. 180 (2009) 1614–1641. `arXiv:0806.4194`, `doi:10.1016/j.cpc.2009.02.018`.

[25] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr, B. Fuks, FeynRules 2.0 - A complete toolbox for tree-level phenomenology`arXiv:1310.1921`.

[26] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, S. Hua-Sheng, P. Torrielli, M. Zaro, aMC@NLO: Automation of next-to-leading order computations in QCD, in preparation.

[27] C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer, et al., UFO - The Universal FeynRules Output, Comput.Phys.Commun. 183 (2012) 1201–1214. arXiv:1108.2040, doi:10.1016/j.cpc.2012.01.022.

[28] G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, et al., Automated One-Loop Calculations with GoSam, Eur.Phys.J. C72 (2012) 1889. arXiv:1111.2034.

[29] J. Bellm, S. Gieseke, D. Grellscheid, A. Papaefstathiou, S. Platzer, et al., Herwig++ 2.7 Release NotearXiv:1310.6877.

[30] B. A. Kniehl, M. Spira, Low-energy theorems in Higgs physics, Z.Phys. C69 (1995) 77–88. arXiv:hep-ph/9505225, doi:10.1007/s002880050007.

[31] M. A. Shifman, A. Vainshtein, M. Voloshin, V. I. Zakharov, Low-Energy Theorems for Higgs Boson Couplings to Photons, Sov.J.Nucl.Phys. 30 (1979) 711–716.

[32] G. F. Giudice, C. Grojean, A. Pomarol, R. Rattazzi, The Strongly-Interacting Light Higgs, JHEP 06 (2007) 045. arXiv:hep-ph/0703164, doi:10.1088/1126-6708/2007/06/045.

[33] N. D. Christensen, P. de Aquino, C. Degrande, C. Duhr, B. Fuks, et al., A Comprehensive approach to new physics simulations, Eur.Phys.J. C71 (2011) 1541. arXiv:0906.2474, doi:10.1140/epjc/s10052-011-1541-5.

[34] N. D. Christensen, P. de Aquino, N. Deutschmann, C. Duhr, B. Fuks, et al., Simulating spin-$\frac{3}{2}$ particles at colliders, Eur.Phys.J. C73 (2013) 2580. arXiv:1308.1668, doi:10.1140/epjc/s10052-013-2580-x.

[35] C. Duhr, B. Fuks, A superspace module for the FeynRules package, Comput. Phys. Commun. 182 (2011) 2404–2426. arXiv:1102.4191, doi:10.1016/j.cpc.2011.06.009.

[36] B. Fuks, Beyond the Minimal Supersymmetric Standard Model: from theory to phenomenology, Int.J.Mod.Phys. A27 (2012) 1230007. arXiv:1202.4769, doi:10.1142/S0217751X12300074.

[37] A. Pukhov, et al., CompHEP: A package for evaluation of Feynman diagrams and integration over multi-particle phase space. User's manual for version 33, arXiv:hep-ph/9908288.

[38] E. Boos, et al., CompHEP 4.4: Automatic computations from Lagrangians to events, Nucl. Instrum. Meth. A534 (2004) 250–259. arXiv:hep-ph/0403113, doi:10.1016/j.nima.2004.07.096.

[39] T. Hahn, M. Perez-Victoria, Automatized one-loop calculations in four and D dimensions, Comput. Phys. Commun. 118 (1999) 153–165. arXiv:hep-ph/9807565, doi:10.1016/S0010-4655(98)00173-8.

[40] T. Hahn, Generating Feynman diagrams and amplitudes with FeynArts 3, Comput. Phys. Commun. 140 (2001) 418–431. arXiv:hep-ph/0012260, doi:10.1016/S0010-4655(01)00290-9.

[41] T. Hahn, A Mathematica interface for FormCalc-generated code, Comput. Phys. Commun. 178 (2008) 217–221. arXiv:hep-ph/0611273, doi:10.1016/j.cpc.2007.09.004.

[42] T. Hahn, FormCalc 6, PoS ACAT08 (2008) 121.

[43] S. Agrawal, T. Hahn, E. Mirabella, FormCalc 7arXiv:1112.0124.

[44] T. Gleisberg, et al., Event generation with SHERPA 1.1, JHEP 02 (2009) 007. arXiv:0811.4622, doi:10.1088/1126-6708/2009/02/007.

[45] M. Moretti, T. Ohl, J. Reuter, O'Mega: An Optimizing matrix element generator, arXiv:hep-ph/0102195.

[46] N. D. Christensen, C. Duhr, B. Fuks, J. Reuter, C. Speckner, Introducing an interface between WHIZARD and FeynRules, Eur.Phys.J. C72 (2012) 1990. arXiv:1010.3251.

[47] A. Alloul, J. D'Hondt, K. De Causmaecker, B. Fuks, M. Rausch de Traubenberg, Automated mass spectrum generation for new physics, Eur.Phys.J. C73 (2013) 2325. arXiv:1301.5932, doi:10.1140/epjc/s10052-013-2325-x.

[48] J. Alwall, et al., New Developments in MadGraph/MadEvent, AIP Conf. Proc. 1078 (2009) 84–89. arXiv:0809.2410, doi:10.1063/1.3052056.

[49] P. Artoisenet, R. Frederix, O. Mattelaer, R. Rietkerk, Automatic spin-entangled decays of heavy resonances in Monte Carlo simulations, JHEP 1303 (2013) 015. arXiv:1212.3460, doi:10.1007/JHEP03(2013)015.

[50] P. Artoisenet, V. Lemaitre, F. Maltoni, O. Mattelaer, Automation of the matrix element reweighting method, JHEP 1012 (2010) 068. arXiv:1007.3300, doi:10.1007/JHEP12(2010)068.

[51] V. Hirschi, R. Frederix, S. Frixione, M. V. Garzelli, F. Maltoni, et al., Automation of one-loop QCD corrections, JHEP 1105 (2011) 044. arXiv:1103.0621, doi:10.1007/JHEP05(2011)044.

[52] R. Frederix, S. Frixione, F. Maltoni, T. Stelzer, Automation of next-to-leading order computations in QCD: The FKS subtraction, JHEP 0910 (2009) 003. arXiv:0908.4272, doi:10.1088/1126-6708/2009/10/003.

[53] E. Byckling, K. Kajantie, N-particle phase space in terms of invariant momentum transfers, Nucl.Phys. B9 (1969) 568–576. doi:10.1016/0550-3213(69)90271-5.

[54] F. James, Monte-Carlo phase space.

[55] B. Allanach, M. Battaglia, G. Blair, M. S. Carena, A. De Roeck, et al., The Snowmass points and slopes: Benchmarks for SUSY searches, Eur.Phys.J. C25 (2002) 113–123. arXiv:hep-ph/0202233, doi:10.1007/s10052-002-0949-3.

[56] G. Bozzi, B. Fuks, B. Herrmann, M. Klasen, Squark and gaugino hadroproduction and decays in non-minimal flavour violating supersymmetry, Nucl.Phys. B787 (2007) 1–54. arXiv:0704.1826, doi:10.1016/j.nuclphysb.2007.05.031.