

Introducing an interface between FeynRules and WHIZARD

Neil D. Christensen^{1,a}, Claude Duhr^{2,b}, Benjamin Fuks^{3,c}, Jürgen Reuter^{4,5,6,d}, Christian Speckner^{5,e}

¹PITTSburgh Particle physics, Astrophysics and Cosmology Center, Department of Physics and Astronomy, University of Pittsburgh, Pittsburgh, PA 15260, USA

²Institute for Particle Physics Phenomenology, University of Durham, Durham, DH1 3LE, UK

³Institut Pluridisciplinaire Hubert Curien/Département Recherches Subatomiques, Université de Strasbourg/CNRS-IN2P3, 23 Rue du Loess, 67037 Strasbourg, France

⁴DESY Theory Group, Notkestraße 85, 22603 Hamburg, Germany

⁵Physikalisches Institut, Albert-Ludwigs-Universität Freiburg, Hermann-Herder-Straße 3, 79104 Freiburg, Germany

⁶School of Physics and Astronomy, University of Edinburgh, JCMB, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK

Received: 16 February 2012 / Revised: 3 April 2012 / Published online: 4 May 2012
© The Author(s) 2012. This article is published with open access at Springerlink.com

Abstract While Monte Carlo event generators like WHIZARD have become indispensable tools in studying the impact of new physics on collider observables over the last decades, the implementation of new models in such packages has remained a rather awkward and error-prone process. Recently, the FEYNRULES package was introduced which greatly simplifies this process by providing a single unified model format from which model implementations for many different Monte Carlo codes can be derived automatically. In this note, we present an interface which extends FEYNRULES to provide this functionality also for the WHIZARD package, thus making WHIZARD's strengths and performance easily available to model builders.

1 Introduction

The Standard Model (SM) of particle physics provides a successful description of all experimental high-energy data to date. However, despite its success, many fundamental questions remain unanswered, such as the origin of electroweak symmetry breaking, the nature of neutrino masses, the large hierarchy between the electroweak and the Planck scales and the origins of dark matter and the cosmological constant. Attempts to address these questions have led to a wide range of new physics theories, most of them predict-

ing new phenomena at the TeV scale. The Large Hadron Collider (LHC) at CERN is currently probing this scale and will hopefully make it possible to discover, constrain and/or exclude some of the theories beyond the Standard Model (BSM).

Monte Carlo simulations of the particle collisions to be observed at the LHC will play a key role in the exploration of the weak scale, both from the theoretical and experimental sides. While proper modeling of the strong interactions, including parton showering, fragmentation and hadronization, is essential in order to achieve a realistic description of the event distributions at the LHC, BSM signals are expected to show up predominantly in the underlying hard interaction. Therefore, a lot of effort has been put into the development of multi-purpose matrix-element and parton-level event generators such as COMPHEP/CALCHEP [1–3], MADGRAPH/MADEVENT [4–8], SHERPA [9] and WHIZARD [10, 11], which allow, at least in principle, the generation of parton-level events for a large class of Lagrangian-based quantum field theory models.

In practice, however, the implementation of a BSM model into any of these tools can be a tedious and error-prone task. Not only might a model contain hundreds, if not thousands, of interaction vertices which need to be encoded in a format suitable for the generator in question, but in addition, each code follows its own format conventions which need to be respected. To improve this situation, a framework based on the FEYNRULES package [12], addressing not only the implementation but also the validation of new physics models in multi-purpose matrix-element and event generators has recently been proposed [13] and its virtue illustrated in the context of the programs COMPHEP/CALCHEP, MADGRAPH/MADEVENT and SHERPA.

^a e-mail: neilc@pitt.edu

^b e-mail: claudeduhr@durham.ac.uk

^c e-mail: benjamin.fuks@iphc.cnrs.fr

^d e-mail: juergen.reuter@desy.de

^e e-mail: christian.speckner@physik.uni-freiburg.de

In this paper we present a new interface from FEYNRULES to the event generator WHIZARD. It follows the same approach as the already existing interfaces, and allows to export any model implemented in FEYNRULES directly to WHIZARD, extending thus the aforementioned framework to the WHIZARD package. The paper is organized as follows: after giving a short overview of the FEYNRULES and WHIZARD packages in Sect. 2, we discuss the usage, features and limitations of the new interface in Sect. 3. Finally, we give a few examples of using the interface in Sect. 4 before concluding in Sect. 5. The Appendix contains an exhaustive list of the interface options (Appendix A) and a selection of numerical results from the interface validation (Appendix B).

2 The WHIZARD and FEYNRULES packages

2.1 FEYNRULES

FEYNRULES is a MATHEMATICA¹ package that allows to derive Feynman rules from any perturbative quantum field theory-based Lagrangian in an automated way. The input provided by the user is threefold and consists of the Lagrangian defining the model, together with the definitions of all the particles and parameters that appear in the model. Up to now, the public release of FEYNRULES supports scalar, vector, fermion (Dirac, Majorana and Weyl [14]) and spin-two fields, as well as Faddeev–Popov ghosts, while recently, superfields have also been included [15]. Once this information is provided, FEYNRULES can perform basic checks on the sanity of the implementation (hermiticity, normalization of the quadratic terms, ...), and finally computes all the interaction vertices associated with the model and store them in an internal format for later processing.

After the Feynman rules have been obtained, FEYNRULES can export the interaction vertices to various matrix-element generators by means of interfaces provided by the package [13]. The current public release contains interfaces to CALCHEP/COMPHEP, FEYNARTS/FORMCALC, GOSAM [16, 17], MADGRAPH/MADEVENT (both 4 and 5), SHERPA and the newly developed interface to WHIZARD which we present in this note.

Since the matrix-element generators very often have color and/or Lorentz structures hardcoded, the different interfaces check whether all the vertices are compliant with the structures supported by the corresponding matrix-element generators, and discard them in the case they are not supported. The output of an interface consists of a set of files organized in a single directory which can be installed into

the relevant matrix-element generator and used as any other built-in models.

2.2 WHIZARD

WHIZARD [11] is a program package for the automatic and efficient generation of unweighted parton-level events for multileg tree-level processes in both the Standard Model and a large number of BSM models. While the original version of WHIZARD (1.xx) was geared towards ILC physics by offering elaborate options for controlling the beam setup, the new version WHIZARD 2.x.x has been redesigned as a hadron collider tool vastly extending but also including the (ILC) features of version 1 (it now also includes both a k_T -ordered and an analytic parton shower [18]). Although many QCD/NLO and SM background physics topics (cf. e.g. [14, 19–21]) have been tackled using the package, its main focus lies on BSM physics and its collider phenomenology (cf. e.g. [32]). Specifically models focusing on alternative scenarios have been incorporated in WHIZARD and independently been tested and validated [22, 24, 27–31], where specifically also new physics effects from higher-dimensional operators have been investigated. The new interface to FEYNRULES described in this paper makes such studies considerably easier.

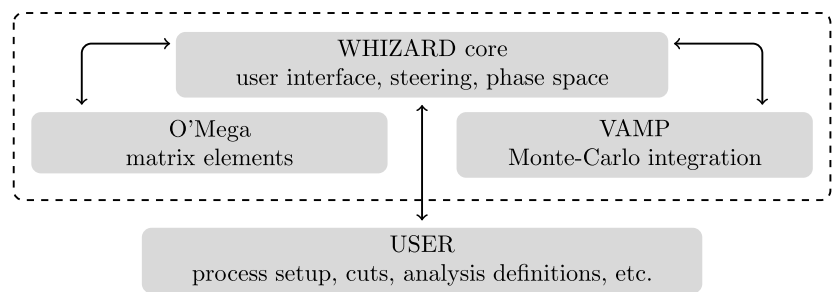
The program is not designed as a monolithic block, but instead shows a high degree of modularity including the, in principle, independent and stand-alone usable matrix-element generator O'MEGA and the adaptive multi-channel Monte Carlo integrator VAMP as well as several dedicated sub-packages for lepton collider physics, all of which are contained in an easy-to-use package. This design is shown schematically in Fig. 1. WHIZARD has been designed such that this structure is completely transparent to the user.

For providing additional functionality like convolution with parton distributions, hadronization or support for additional event formats, WHIZARD can be linked against external packages like LHAPDF [36], PYTHIA [37] or also HEPMC [38].

O'MEGA O'Mega [10] is a generator for tree-level matrix elements which generates a symbolic representation of the scattering amplitude and translates it into FORTRAN 90 code. As this is a computer-algebraic process which is rather similar to the workflow of an optimizing compiler, the OCAML language has been chosen for the implementation of O'MEGA.

The algorithm is based on the algebraic paradigm of so-called Directed Acyclical Graphs (DAGs) which allow a re-usage of all already computed components. Inspired by the ideas of HELAS [39], amplitudes are constructed by building a representation of the matrix element as a tree of one-particle irreducible wave functions (IPOWs, Greens functions with all but one leg amputated) which is then translated into highly optimized FORTRAN 90 code to evaluate

¹ MATHEMATICA is a registered trademark of Wolfram Research Inc.

Fig. 1 General structure of the WHIZARD package

the matrix element by numerically fusing IPOWs. This approach avoids the repeated and redundant evaluation of sub-diagrams present in any Feynman diagram based approach and brings the factorial growth in complexity of the latter down to an exponential behavior. In addition, the optimized structure of the amplitudes is especially well-suited to cope with big cancellations present in theories with gauge invariances.

The algebraic O'MEGA algorithm bears similarities to the numerical recursive solution of Schwinger-Dyson equations of motion implemented in the code ALPHA [40]. In this program, however, the evaluation proceeds fully numerically and no symbolic representation of the amplitude is constructed.

While there is no design limitation on the spin of the fields or on their interactions, O'MEGA currently supports spin 0, $\frac{1}{2}$, 1, $\frac{3}{2}$ and 2 fields, most dimension three and four couplings between those (in particular all dimension four gauge interactions) and a small set of higher dimension operators.²

Color treatment in O'MEGA leverages the color flow [41, 42] decomposition and currently supports fields transforming either as singlets, triplets, antitriplets or octets. Only the vertex factor belonging to a predefined flow is implemented in the model definition, and the other flows are derived by O'MEGA from the $SU(3)_C$ representations of the fields.

Models are represented in O'MEGA as OCAML modules which contain symbolic definitions of the fields and coupling constants, vertex lists, translation functions which map the fields onto their Lorentz/ $SU(3)_C$ representations and a number of auxiliary maps which provide textual representations of the fields and constants for interfacing and code generation purposes. These modules are compiled and linked with the O'MEGA framework to obtain executables which can be called to transform a process definition into FORTRAN code.

²A future revision of WHIZARD/O'MEGA will allow specifying the Lorentz structure of the couplings directly in the model file, thus alleviating the limitation to predefined structures and permitting a much larger set of couplings.

VAMP VAMP [43] is a multichannel extension of the VEGAS [44] algorithm. For all possible singularities in the integrand, suitable maps and integration channels are chosen which are then weighted and superimposed to build the phase space parameterization. Both grids and weights are modified in the adaption phase of the integration.

The multichannel integration algorithm is implemented as a FORTRAN 90 library with the task of mapping out the integrand and finding suitable parameterizations being completely delegated to the calling program (WHIZARD core in this case). This makes the actual VAMP library completely agnostic of the model under consideration.

WHIZARD core With matrix element generation and Monte Carlo integration being outsourced into dedicated building blocks, it is the job of the program core to roll O'MEGA and VAMP transparently into a package which can be operated without dealing with the components directly.

To this end, the core provides a command-line based user interface which is used to specify the process, event generation and analysis setup and to control the runs of the event generator. In addition, it is responsible for all physics pieces not covered by the other two components, most importantly phase space generation and beam (parton densities) setup.

The major task of the WHIZARD core is to generate the phase space and map out the integrand, for which it needs information on the model under consideration. In the legacy 1.x branch of WHIZARD, the core is implemented as a set of FORTRAN 90 modules, PERL glue for managing the code generation and Makefiles for steering the build process. New models have to be added to the build framework in order to be available to the user.

The recently published 2.x branch of WHIZARD supersedes the old versions and features a complete rewriting of the core as a single FORTRAN 2003 program, discarding the old PERL code. Models are loaded dynamically and searched for in a configurable search path, allowing the user to add new models without modifying the WHIZARD package itself. Moreover, the dedicated scripting language SINDARIN has been introduced which steers all aspects of the simulation and which replaces the set of input files used in previous versions.

3 Implementing new models in WHIZARD using FEYNRULES

3.1 General strategy

In this section, we describe how to implement generic BSM models into WHIZARD in an efficient way by means of the newly developed interface from FEYNRULES to WHIZARD.

This new interface brings the implementation of new models into WHIZARD in line with the approach introduced in Ref. [13], where its power to develop, implement and validate models has been demonstrated in the context of CALCHEP/COMPHEP, MADGRAPH/MADEVENT and SHERPA. This approach is built around FEYNRULES, a MATHEMATICA package where any perturbative and local quantum field theory Lagrangian can be implemented and its Feynman rules obtained in an automated way. The interaction vertices can then be passed, together with all relevant information, to various matrix element generators through dedicated interfaces.

The new FEYNRULES interface to WHIZARD allows the user to integrate any model implemented in FEYNRULES into the WHIZARD/O'MEGA framework, although some technical restrictions apply and will be explained in the next section. The interface produces a set of files that contain all the particle and parameter definitions of the model together with the interaction vertices. Together with the model files, a framework is created which allows to communicate the new models to WHIZARD in a well defined way, after which step the model can be used exactly like the built-in ones. This specifically means that the user is not required to manually

modify the code of WHIZARD/O'MEGA, the models created by the interface can be used directly without any further user intervention.

3.2 Features and restrictions of the WHIZARD–FEYNRULES interface

In this section we describe in detail the features and restrictions of the new interface from FEYNRULES to WHIZARD. After initializing FEYNRULES and loading a model definition, the code can be invoked in a similar fashion as the other interfaces described in Ref. [13] in order to transform the Feynman rules obtained automatically from a Lagrangian into a set of model files ready to use with WHIZARD. Both the legacy branch 1.9x as well as the current 2.x series of WHIZARD are actively supported by the code generator. Options that can be passed to the interface are introduced only as needed, while a detailed explanation of all options can be found in Appendix A.

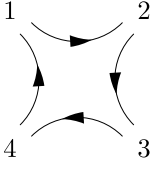
Supported fields and vertices All fields currently supported by FEYNRULES (scalars, Dirac and Majorana fermions, vectors and symmetric tensors) are handled by the interface. The set of accepted operators, the full list of which can be found in Table 1, is a subset of all the operators supported by O'MEGA. While still limited, this list is sufficient for a large number of BSM models. In addition, a future version of WHIZARD/O'MEGA will support the definition of completely general Lorentz structures in the model, allowing the interface to translate all interactions handled by FEYNRULES.

Table 1 All Lorentz structures currently supported by the WHIZARD–FEYNRULES interface, sorted with respect to the spins of the particles

Particle spins	Supported Lorentz structures
FFS	All operators of dimension four are supported
FFV	All operators of dimension four are supported
SSS	All dimension three interactions are supported
SVV	Supported operators: dimension 3: $\mathcal{O}_3 = V_1^\mu V_{2\mu} \phi$ dimension 5: $\mathcal{O}_5 = \phi (\partial^\mu V_1^\nu - \partial^\nu V_1^\mu) (\partial_\mu V_{2\nu} - \partial_\nu V_{2\mu})$ Note that \mathcal{O}_5 generates the effective gluon-gluon-Higgs couplings obtained by integrating out heavy quarks
SSV	$(\phi_1 \partial^\mu \phi_2 - \phi_2 \partial^\mu \phi_1) V_\mu$ type interactions are supported
SSVV	All dimension four interactions are supported
SSSS	All dimension four interactions are supported
VVV	All parity-conserving dimension four operators are supported, with the restriction that non-gauge interactions may be split into several vertices and can only be handled if all three fields are mutually different
VVVV	All parity conserving dimension four operators are supported
TSS, TVV, TFF	The three point couplings in the Appendix of Ref. [45] are supported

Notes: “S” stands for scalar, “F” for fermion (either Majorana or Dirac) and “V” for vector

Table 2 All possible combinations of three or four $SU(3)_C$ representations supported by FEYNRULES which can be used to build singlets, together with the support status of the corresponding color structures in WHIZARD and the interface

$SU(3)_C$ representations	Support status
111, $\bar{3}31$, $\bar{3}38$, 1111, $\bar{3}311$, $\bar{3}381$	Fully supported by the interface
888, 8881	Supported only if at least two of the octets are identical particles
881, 8811	Supported in output for WHIZARD 1.96 and higher (including the new 2.x branch) only ^a
$\bar{3}388$	Supported only if the octets are identical particles
8888	The only supported flow structure is
	
	where $\Gamma(1, 2, 3, 4)$ represents the Lorentz structure associated with the first flow
333, $\bar{3}\bar{3}\bar{3}$, 3331, $\bar{3}\bar{3}\bar{3}1$, $\bar{3}\bar{3}33$	Unsupported

^aNote that in order to use such couplings in the 1.9x branch, the O'Mega option 2g must be added to the process definition in whizard.prc

Color Color is treated in O'MEGA in the color flow decomposition, with the flow structure being implicitly determined from the representations of the particles present at the vertex. Therefore, the interface has to strip the color structure from the vertices derived by FEYNRULES before writing them out to the model files. While this process is straightforward for all color structures which correspond only to a single flow assignment, vertices with several possible flow configurations must be treated with care in order to avoid mismatches between the flows assigned by O'MEGA and those actually encoded in the couplings. To this end, the interface derives the color flow decomposition from the color structure determined by FEYNRULES and rejects all vertices which would lead to a wrong flow assignment by O'MEGA (these rejections are accompanied by warnings from the interface).

At the moment, the $SU(3)_C$ representations supported by both WHIZARD and the interface are singlets (1), triplets (3), antitriplets ($\bar{3}$) and octets (8). Table 2 shows all combinations of these representations which can form singlets together with the support status of the respective color structures in WHIZARD and the interface. Although the supported color structures do not comprise all possible singlets, the list is sufficient for a large number of SM extensions. Furthermore, a future revision of WHIZARD/O'MEGA will allow for explicit color flow assignments, thus removing most of the current restrictions.

In models generated for the 1.9x branch of WHIZARD, a hardcoded limit exists on the maximum number n_{cf} of external color lines which is related to the number of external

octets n_8 and triplets/antitriplets n_3 as

$$n_{cf} = n_8 + \frac{n_3}{2}.$$

This limit is set to $n_{cf} = 4$ by default and can be changed via the option WOMaxNcf. No such limit applies to WHIZARD 2.0.0 and higher.

Running α_S Starting with version 2.0, WHIZARD supports a running strong coupling. While this is fully supported by the interface, a choice has to be made which quantities are to be reevaluated when the strong coupling is evolved. We choose to implement the running such that by default aS, G (see Ref. [12] for the nomenclature regarding the QCD coupling) and any vertex factors depending on them are evolved. The list of internal parameters that are to be recalculated (together with the vertex factors depending on them) can be extended (beyond aS and G) by using the option WORunParameters when calling the interface. As older versions of WHIZARD do not evolve the strong coupling, the interface does not support running with WHIZARD 1.x.

Gauge choices The interface supports the unitarity, Feynman and R_ξ gauges. The choice of gauge must be communicated to the interface via the option WOGauge. Note that massless gauge bosons are always treated in Feynman gauge.

If the selected gauge is Feynman or R_ξ , the interface can automatically assign the proper masses to the Goldstone bosons. This behavior is requested by using the WOAUTO-Gauge option. In the R_ξ gauges, the symbol representing

the gauge ξ must be communicated to the interface by using the `WOGaugeSymbol` option (the symbol is automatically introduced into the list of external parameters if `WOAutoGauge` is selected at the same time). This feature can be used to automatically extend models implemented in Feynman gauge to the R_ξ gauges.

Since WHIZARD is a tree-level tool working with helicity amplitudes, the ghost sector is irrelevant for WHIZARD and can be left out. Ghosts and interactions involving ghosts are dropped by the interface.

3.3 Usage

Installation and basic usage From version 1.6.0 onward, the interface is included into the official FEYNRULES distribution. In addition, the latest version can be downloaded from the WHIZARD homepage on HEPFORGE. An installer is included which installs the interface into an existing FEYNRULES installation (this allows to use the program with FEYNRULES 1.4.x where it is not part of the package).

Once installed, the interface can be called and used in the same way as the other interfaces described in Ref. [12]. For example, once the FEYNRULES environment has been initialized and a model has been loaded, the command

```
WriteWOOutput[L]
```

will call `FeynmanRules` to extract the Feynman rules from the Lagrangian `L`, translate them together with the model data and finally write the files necessary for using the model with WHIZARD to an output directory (the name of which is inferred from the model name by default). Options can be added for further control over the translation process (see Appendix A). Instead of using a Lagrangian, it is also possible to call the interface on a vertex list. For example, the following command

```
WriteWOOutput[Input -> list]
```

will directly translate the vertex list `list`. Note that this vertex list must be given in flavor-expanded form in order for the interface to process it correctly.

The interface also supports the `WriteWOExtParams` command described in Ref. [12] for the other interfaces. Issuing

```
WriteWOExtParams[filename]
```

will write a list of all the external parameters to `filename`. The format of the output depends on the WHIZARD version it is targeted at: for 1.9x, it is a FORTRAN namelist suitable for use in `whizard.in`, while a SINDARIN script is generated for 2.x. The target version is selected with the option `WOWhizardVersion` (which is the only option accepted by the command).

During execution, the interface will print out a series of messages. It is highly advised to carefully read through this

output as it not only summarizes the settings and the location of the output files, but also contains information on any skipped vertices or potential incompatibilities of the model with WHIZARD.

Propagating the output to WHIZARD After the interface has run successfully and written the model files to the output directory, the model must be imported into WHIZARD. This step depends on whether the code has been generated for the legacy 1.9x branch or for 2.x.

In the case of version 1.9x, the model files must be patched into the WHIZARD tree before the model can be used. For this purpose, a script called `inject.sh` is created in the output directory. For most applications, this step is as easy as issuing (assuming that the output directory is the current working directory)

```
./inject.sh /path/to/whizard
```

(replacing the path with the correct path to the WHIZARD installation). After reconfiguring and recompiling WHIZARD, the model is available and can be used in a similar fashion as the built-in ones. The installer supports several options for more complex usage scenarios which are described in the `INSTALL` created in the output directory.

For the new 2.x branch, the model files are compiled and can then be installed independently of WHIZARD. In the simplest scenario, assuming that the output directory is the current working directory and that the WHIZARD binaries can be found in the current `$(PATH)`, the installation is performed by issuing

```
./configure && make clean && make  
install
```

This will compile the model and install it into the directory `$(HOME)/.whizard`, making it fully available to WHIZARD without any further intervention. The build system can be adapted to more complicated cases through several options to `configure` which are listed in the `INSTALL` file created in the output directory.

3.4 Validation of the interface

The output of the interface has been extensively validated for both the 1.9x and 2.x branches. Specifically, the integrated cross sections for all possible $2 \rightarrow 2$ processes in the FEYNRULES SM, the MSSM and the Three-Site Higgsless Model [46] have been compared between WHIZARD, MADGRAPH and CALCHEP (for all three of which interfaces are included in FEYNRULES). In addition, the native versions of these models for the respective codes have also been included into the comparison.³ For those codes which support it (CALCHEP and WHIZARD), different gauges have

³A native version of the Three-Site Model is available only for Calchep and WHIZARD.

been checked as well. In all comparisons, excellent agreement within the Monte Carlo errors was achieved. Several tables showcasing the results for selected processes in all three models can be found in Appendix B.

4 Usage examples

In order to illustrate the usage of the interface as detailed above, we will now show several examples of generating output for WHIZARD 2. The examples are constructed to show the application of the different options of the interface and to serve as a starting point for the generation of WHIZARD versions of other FEYNRULES models.

4.1 Standard Model

To start off, we will create WHIZARD 2 versions of the Standard Model as implemented in FEYNRULES for different gauge choices.

Unitarity gauge In order to invoke FEYNRULES, we change to the corresponding directory and load the program in MATHEMATICA via

```
$FeynRulesPath =
  SetDirectory["<path-to-FeynRules>"];
<<FeynRules`
```

The model is loaded by

```
LoadModel["Models/SM/SM.fr"];
FeynmanGauge = False;
```

Note that the second line is required to switch the Standard Model to Unitarity gauge as opposed to Feynman gauge (which is the default). Generating a WHIZARD 2 version of the model is now as easy as doing

```
WriteWOOutput[LSM];
```

After invocation, the interface first gives a short summary of the setup

```
Short model name is "fr_standard_model"
Gauge: Unitarity
Generating code for WHIZARD/O'Mega
          version 2.0.3
Maximum number of couplings per FORTRAN
          module: 500
Extensive Lorentz structure checks
disabled.
```

Note that, as we have not changed any options, those settings represent the defaults. The output proceeds with the calculation of the Feynman rules from the Standard Model Lagrangian LSM. After the rules have been derived, the interface starts generating output and tries to match the vertices to their WHIZARD/O'MEGA counterparts.

```
10 of 75 vertices processed...
20 of 75 vertices processed...
30 of 75 vertices processed...
40 of 75 vertices processed...
50 of 75 vertices processed...
60 of 75 vertices processed...
70 of 75 vertices processed...
processed a total of 75 vertices, kept 74
of them and threw away 1, 1 of which
contained ghosts or goldstone bosons.
```

The last line of the above output is particularly interesting, as it informs us that everything worked out correctly: the interface was able to match all vertices, and the only discarded vertex was the QCD ghost interaction. After the interface has finished running, the model files in the output directory are ready to use and can be compiled using the procedure described in the previous section.

Feynman and R_ξ gauges As the Standard Model as implemented in FEYNRULES also supports Feynman gauge, we can use the program to generate a Feynman gauge version of the model. Loading FEYNRULES and the model proceeds as above, with the only difference being the change

```
FeynmanGauge = True;
```

In order to inform the interface about the modified gauge, we have to add the option WOGauge

```
WriteWOOutput[LSM, WOGauge -> WOFeynman];
```

The modified gauge is reflected in the output of the interface

```
Short model name is "fr_standard_model"
Gauge: Feynman
Generating code for WHIZARD/O'Mega
          version 2.0.3
Maximum number of couplings per FORTRAN
          module: 500
Extensive Lorentz structure checks
disabled.
```

The summary of the vertex identification now takes the following form

```
processed a total of 163 vertices, kept
139 of them and threw away 24, 24 of
which contained ghosts.
```

Again, this line tells us that there were no problems—the only discarded interactions involved the ghost sector which is irrelevant for WHIZARD.

As WHIZARD does not depend on the ghost sector, the only difference between the different gauges from the perspective of the interface are the gauge boson propagators and the Goldstone boson masses. Therefore, the interface can automatically convert a model in Feynman gauge to a model in R_ξ gauge. To this end, the call to the interface must be changed to

```
WriteWOOutput[LSM, WOGauge -> WORxi,
              WOAutoGauge -> True];
```

The `WOAutoGauge` argument instructs the interface to automatically

1. Introduce a symbol for the gauge parameter ξ into the list of external parameters
2. Generate the Goldstone boson masses from those of the associated gauge bosons (ignoring the values provided by FEYNRULES)

The modified setup is again reflected in the interface output

```
Short model name is "fr_standard_model"
Gauge: Rxi
Gauge symbol: "Rxi"
Generating code for WHIZARD/O'Mega
                    version 2.0.3
Maximum number of couplings per FORTRAN
                        module: 500
Extensive Lorentz structure checks
    disabled.
```

Note the default choice `Rxi` for the name of the ξ parameter—this can be modified via the option `WOGaugeParameter`.

While the `WOAutoGauge` feature allows to generate R_ξ gauged models from models implemented in Feynman gauge, it is of course also possible to use models genuinely implemented in R_ξ gauge by setting this parameter to `False`. Also, note that the choice of gauge only affects the propagators of massive fields. Massless gauge bosons are always treated in Feynman gauge.

Compilation and usage In order to compile and use the freshly generated model files, change to the output directory which can be determined from the interface output (in this example, it is `fr_standard_model-WO`). Assuming that WHIZARD is available in the binary search path, compilation and installation proceeds as described in the previous section by issuing (from the shell)

```
./configure && make && make install
```

The model is now ready and can be used similarly to the builtin WHIZARD models. For example, a minimal WHIZARD input file for calculating the $e^+e^- \rightarrow W^+W^-$ scattering cross section in the freshly generated model would look like

```
model = fr_standard_model
process test = "e+", "e-" -> "W+", "W-"
sqrts = 500 GeV
integrate (test)
```

4.2 Minimal supersymmetric Standard Model

In this section, we illustrate the usage of the interface between FEYNRULES and WHIZARD in the context of the

MSSM. For the conventions on the names of the particles and the relative signs entering the different terms of the Lagrangian, we refer to Refs. [13, 15]. All the parameters of the model are then ordered in Les Houches blocks and counters following the SUSY Les Houches Accord (SLHA) [47, 48]. Let us emphasize that implementing any model in FEYNRULES only requires a SLHA-like structure, whilst the choice of the SLHA conventions for the MSSM is only due to the author of the model implementation.

The neutralino sector deserves special attention. After diagonalization of the mass matrix expressed in terms of the gaugino and higgsino eigenstates, the resulting mass eigenvalues may be either negative or positive. In this case, two procedures can be followed. Either the masses are rendered positive and the associated mixing matrix gets purely imaginary entries or the masses are kept signed, the mixing matrix in this case being real. According to the SLHA agreement, the second option is adopted. For a specific eigenvalue, the phase is absorbed into the definition of the relevant eigenvector, rendering the mass negative. However, WHIZARD does not officially support negative masses. For the case of the interface to FEYNRULES this means, that one must be careful using a SLHA file with explicit factors of the complex unity in the mixing matrix, and on the other hand, real and positive masses for the neutralinos. For the hard-coded SUSY models, this is completely handled internally, and the SUSY implementations in WHIZARD have been extensively tested [23, 25, 26, 33–35]. Especially Ref. [33] discusses the details of the neutralino (and chargino) mixing matrix.

After having downloaded the model from the FEYNRULES website, we store it in a new directory, labeled MSSM, of the model library of the local installation of FEYNRULES. The model can then be loaded in MATHEMATICA as

```
$FeynRulesPath =
  SetDirectory["<path-to-FeynRules>"];
<<FeynRules`
LoadModel["Models/MSSM/MSSM.fr"];
FeynmanGauge = False;
```

As it can be seen from the last command line, we have adopted the unitarity gauge. Other choices, such as Feynman gauge, are also possible, and we refer to Sect. 4.1 for using the interface with other gauge choice.

The number of vertices associated to supersymmetric Lagrangians is in general very large (several thousands). For such models with many interactions, it is recommended to first extract all the Feynman rules of the theory before calling the interface between FEYNRULES and WHIZARD. The reason is related to the efficiency of the interface which takes a lot of time in the extraction of the interaction vertices. In the case one wishes to study the phenomenology of several benchmark scenarios, this procedure, which is illustrated below, allows to use the interface in the most

optimal way. The Feynman rules are derived from the Lagrangian once and for all and then reused by the interface for each set of WHIZARD model files to be produced, considerably speeding up the generation of multiple model files issued from a single Lagrangian. In addition, the scalar potential of supersymmetric theories contains a large set of four scalar interactions, in general irrelevant for collider phenomenology. These vertices can be neglected with the help of the `Exclude4Scalars->True` option of both the commands `FeynmanRules` and `WriteWOOutput`. The Feynman rules of the MSSM are then computed by issuing in the MATHEMATICA notebook,

```
rules = FeynmanRules[lag,
  Exclude4Scalars->True,
  FlavorExpand->True];
```

where `lag` is the variable containing the Lagrangian.

By default, all the parameters of the model are set to the value of 1. A complete parameter file must therefore be loaded. We choose the `SPS1a`⁴ benchmark point [49], where we take care of setting all the masses of the neutralinos to positive values and adapt the associated mixing matrix accordingly. This parameter file, named `sps1a_wo.dat`, can be downloaded from the FEYNRULES website, and loaded into FEYNRULES as

```
ReadLHAFile[Input -> "sps1a_wo.dat"];
```

We note that this command does not reduce the size of the model output by removing vertices with vanishing couplings. However, if desired, this task could be done with the `LoadRestriction` command (see Ref. [50] for details).

The vertices are now ready to be exported to WHIZARD with the help of the command

```
WriteWOOutput[Input -> rules];
```

For compilation and usage of the produced model, we refer to Sect. 4.1. Finally, let us note that the numerical values of the parameters of the model can be modified directly at the level of the WHIZARD program, without having to generate a second time the WHIZARD model files from FEYNRULES. To this end, a SINDARIN script is created by FEYNRULES with the help of the instruction

```
WriteWOExtParams["parameters.sin"];
```

and can be further modified according to the needs of the user when employing the WHIZARD program.

4.3 Three-site Higgsless model

The Three-Site or Minimal Higgsless Model (MHM) [46] is a minimal deconstructed Higgsless model which contains only the first resonance in the tower of Kaluza–Klein

modes of a Higgsless extra-dimensional model. It is a non-renormalizable, effective theory whose gauge group is an extension of the SM with an extra $SU(2)$ gauge group. The breaking of the extended electroweak gauge symmetry is accomplished by a set of nonlinear sigma fields which represent the effects of physics at a higher scale and make the theory nonrenormalizable. The physical vector boson spectrum contains the usual photon, W^\pm and Z bosons as well as a W'^\pm and Z' boson. Additionally, a new set of heavy fermions are introduced to accompany the new gauge group “site” which mix to form the physical eigenstates. This mixing is controlled by the small mixing parameter ϵ_L which is adjusted to satisfy constraints from precision observables, such as the S parameter [51].

The MHM has been implemented into LANHEP [52], FEYNRULES [13] and independently into WHIZARD [53], and the collider phenomenology has been studied by making use of these implementations [22, 52, 53]. Furthermore, the independent implementations in FEYNRULES and directly into WHIZARD have been compared and found to agree (see Appendix B). In this section, we describe the use of the WHIZARD interface to generate WHIZARD model files for the MHM. This model has been implemented in Feynman gauge as well as unitarity gauge and contains the variable `FeynmanGauge` which can be set to `True` or `False`. When set to `True`, the option `WOGauge-> WOFeynman` must be used, as described in Appendix A. R_ξ gauge can also be accomplished with this model by use of the options `WOGauge -> WORxi` and `WOAutoGauge -> True`.

Since this model makes use of the nonlinear sigma field

$$\Sigma = 1 + i\pi - \frac{1}{2}\pi^2 + \dots \quad (1)$$

many higher dimensional operators are included in the model which are not supported by WHIZARD. Although WHIZARD can reject these vertices and print a warning message to the user, it is preferable to remove the vertices with the option `MaxCanonicalDimension->4` which is passed to `FeynmanRules` and restricts the Feynman rules to those of dimension four and smaller.⁵

Since the use of different gauges was already illustrated in Sect. 4.1, we will simply make use of this interface to output the model in Feynman gauge. We load FEYNRULES with

```
$FeynRulesPath =
  SetDirectory["<path-to-FeynRules>"];
<<FeynRules`
```

⁴Though we are aware that this specific point is excluded by LHC data, we use it here purely for demonstration purposes, as the validation of the MSSM within the interface has been done with it.

⁵We note, at this stage, that `MaxCanonicalDimension` is an option of the `FeynmanRules` function rather than of the interface, itself. In fact, the interface accepts all the options of `FeynmanRules` and simply passes them on to the latter.

where `<path-to-FeynRules>` is the path to the FEYN-RULES files. The MHM is loaded by

```
SetDirectory["<path-to-MHM>"];
LoadModel["3-Site-particles.fr",
  "3-Site-parameters.fr",
  "3-Site-lagrangian.fr"];
FeynmanGauge = True;
```

where `<path-to-MHM>` is the path to the directory where the MHM model files are stored and where the output of the WHIZARD interface will be written. The WHIZARD interface is initiated with the command

```
WriteWOutput[LGauge, LGold, LGhost,
  LFermion, LGoldLeptons, LGoldQuarks,
  MaxCanonicalDimension->4,
  WOGauge->WOFeynman,
  WOModelName->"fr_mhm"];
```

where we have also made use of the option `WOModelName` to change the name of the model as seen by WHIZARD (see Appendix A). As in the SM (see Sect. 4.1), the interface begins by writing a short informational message:

```
Short model name is "fr_mhm"
Gauge: Feynman
Generating code for WHIZARD/O'Mega
                        version 2.0.3
Automagically assigning Goldstone
                        boson masses...
Maximum number of couplings per FORTRAN
                        module: 500
Extensive Lorentz structure checks
disabled.
```

After calculating the Feynman rules and processing the vertices, the interface gives the summary:

```
processed a total of 922 vertices, kept
  633 of them and threw away 289, 289 of
  which contained ghosts.
```

showing that no vertices were missed. The files are stored in the directory `fr_mhm` and are ready to be installed and used with WHIZARD.

5 Conclusions

In this paper, we presented the interface between FEYN-RULES, a program to automatically generate Feynman rules from a generic input Lagrangian, to the multi-particle event generator WHIZARD. The interface can be used both for the legacy version WHIZARD 1.xx as well as the actual release line WHIZARD 2.x and is included in FEYNRULES from version 1.6 on. There has been a quite exhaustive validation of the interface both with FEYNRULES models used with other Monte Carlo generators as well as with hard-coded implementations of models. We tried to keep the description of

the interface rather self-contained, but recommend the user to consult both WHIZARD and FEYNRULES manuals when performing a calculation with the interface. The paper gives a complete overview of the supported particle types as well as the Lorentz and color structures of interaction vertices. We listed the options and flags which steer the compilation of physics models and the transferring of the Feynman rules into the generator. In order to give examples of the workflow, we applied it to the three models used in our validation procedure. From the steps given in the paper to go from the Lagrangian formulation of that model through FeynRules to a simulation within Whizard, the user can easily apply the interface to his or her own preferred model. The presented FEYNRULES/WHIZARD interface hopefully facilitates phenomenological studies at present and future collider experiments. A prime example of using the combination between FEYNRULES and an event generator like WHIZARD is to investigate models whose particle content depend on the point in parameter space, which is however a separate project and will be presented elsewhere [54].

Acknowledgements C.S. has been supported by the Deutsche Forschungsgemeinschaft through the Research Training Groups GRK 1147 *Theoretical Astrophysics and Particle Physics* and GRK 1102 *Physics of Hadron Accelerators*. J.R.R. has been partially supported by the Ministry of Science and Culture (MWK) of the German state Baden-Württemberg. B.F. acknowledges support by the Theory-LHC France-initiative of the CNRS/IN2P3. N.D.C. was supported by the US National Science Foundation under grants PHY-0354226 and PHY-0705682 and by the Pittsburgh Particle physics, Astrophysics and Cosmology Center. CS would like to thank the Michigan State University and especially S. Chivukula and L. Simmons for their hospitality and T. Ohl for encouraging him to start this project.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Appendix A: Options of the WHIZARD interface

In the following we present a comprehensive list of all the options accepted by `WriteWOutput`. Additionally, we note that all options of FeynmanRules are accepted by `WriteWOutput`, which passes them on to FeynmanRules.

Input

An optional vertex list to use instead of a Lagrangian (which can then be omitted).

WOWhizardVersion

Select the WHIZARD version for which code is to be generated. The currently available choices are summarized in Table 3. This list will expand as the program evolves. To get a summary of all choices available in a particular version of the interface, use the command `?WOWhizardVersion`.

Table 3 Currently available choices for the WWhizardVersion option, together with the respective WHIZARD versions supported by them

WWhizardVersion	WHIZARD versions supported
"1.92"	1.92
"1.93"	1.93–1.95
"1.96"	1.96+
"2.0"	2.0.0–2.0.2
"2.0.3" (default)	2.0.3+

WOModelName

The name under which the model will be known to WHIZARD. Please note that models for versions 1.9x must start with “fr_” if they are to be picked up by WHIZARD automatically. The default is determined from the FEYNRULES model name.

Output

The name of the output directory. The default is determined from the FEYNRULES model name.

Table 4 Cross sections for a selection of two-to-two processes in the SM

Process	\sqrt{s} [GeV]	FEYNRULES							Stock		
		CH	CH	MG	WO1	WO1	WO2	WO2	CH	MG	WO2
		F	U	U	F	U	F	U	U	U	U
$W^+ W^+ \rightarrow W^+ W^+$	1277	25.7	25.7	25.7	25.7	25.7	25.7	25.7	25.7	25.7	25.7
$g g \rightarrow g g$	200	1.88e04	1.88e04	1.89e04	1.88e04	1.88e04	1.88e04	1.88e04	1.88e04	1.88e04	1.88e04
$Z Z \rightarrow W^+ W^-$	1368	26.2	26.2	26.2	26.2	26.2	26.2	26.2	26.2	26.2	26.2
$\gamma \gamma \rightarrow W^+ W^-$	639	16.3	16.3	16.3	16.3	16.3	16.3	16.3	16.3	16.3	16.3
$Z Z \rightarrow Z Z$	1459	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245	0.245
$\gamma Z \rightarrow W^+ W^-$	1003	19.3	19.3	19.3	19.3	19.3	19.3	19.3	19.3	19.3	19.3
$s \bar{s} \rightarrow Z Z$	730	0.0838	0.0838	0.0839	0.0838	0.0838	0.0838	0.0838	0.0838	0.0839	0.0838
$\mu^- \mu^+ \rightarrow \gamma Z$	366	1.46	1.46	1.46	1.46	1.46	1.46	1.46	1.46	1.46	1.46
$s \bar{s} \rightarrow \gamma \gamma$	200	0.0272	0.0272	0.0272	0.0272	0.0272	0.0272	0.0272	0.0272	0.0272	0.0272
$u \bar{d} \rightarrow Z W^+$	684	0.146	0.146	0.146	0.146	0.146	0.146	0.146	0.146	0.146	0.146
$c \bar{c} \rightarrow g g$	200	462	462	462	462	462	462	462	462	462	462
$\nu_e e^+ \rightarrow \gamma W^+$	319	1.98	1.98	1.98	1.98	1.98	1.98	1.99	1.98	1.98	1.98
$d \bar{d} \rightarrow \gamma Z$	365	0.0798	0.0798	0.0799	0.0798	0.0798	0.0798	0.0798	0.0798	0.0798	0.0798
$b \bar{b} \rightarrow Z Z$	767	0.0760	0.0760	0.0758	0.0760	0.0760	0.0760	0.0760	0.0760	0.0758	0.0760
$c \bar{s} \rightarrow \gamma W^+$	325	0.208	0.208	0.208	0.207	0.208	0.208	0.208	0.208	0.208	0.208
$\nu_\tau \bar{\nu}_\tau \rightarrow W^+ W^-$	639	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06	1.06
$\nu_\mu \bar{\nu}_\tau \rightarrow \mu^- \tau^+$	200	16.5	16.5	16.5	16.5	16.5	16.5	16.5	16.5	16.5	16.5
$\nu_\mu \bar{\nu}_\mu \rightarrow \mu^- \mu^+$	200	22.4	22.4	22.4	22.4	22.4	22.4	22.4	22.4	22.4	22.4
$t \bar{t} \rightarrow s \bar{s}$	1395	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14	1.14
$e^- e^+ \rightarrow t \bar{t}$	1394	0.0723	0.0723	0.0724	0.0723	0.0723	0.0723	0.0723	0.0723	0.0723	0.0723
$c c \rightarrow c c$	200	3.41e03	3.41e03	3.42e03	3.41e03	3.41e03	3.42e03	3.41e03	3.41e03	3.41e03	3.42e03
$u \bar{u} \rightarrow s \bar{s}$	200	80.3	80.3	80.2	80.3	80.3	80.3	80.3	80.3	80.3	80.3
$u \bar{u} \rightarrow d \bar{d}$	200	68.1	68.1	68.0	68.1	68.1	68.1	68.1	68.1	68.1	68.0
$u \bar{t} \rightarrow d \bar{b}$	716	5.49	5.49	5.49	5.49	5.49	5.49	5.49	5.49	5.50	5.49
$\nu_\tau \tau^+ \rightarrow u \bar{s}$	200	0.501	0.501	0.501	0.501	0.501	0.501	0.501	0.501	0.501	0.501
$\nu_\mu \mu^+ \rightarrow c \bar{s}$	200	9.33	9.33	9.32	9.33	9.33	9.33	9.33	9.33	9.33	9.33
$Z Z \rightarrow H H$	1690	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250
$W^+ W^- \rightarrow H H$	1599	0.151	0.151	0.151	0.151	0.151	0.151	0.151	0.151	0.151	0.151
$t \bar{t} \rightarrow H H$	2354	0.0593	0.0593	0.0593	0.0593	0.0593	0.0593	0.0593	0.0593	0.0593	0.0593
$\tau^- \tau^+ \rightarrow H H$	974	2.96e-06	2.96e-06	2.96e-06	2.96e-06	2.96e-06	2.96e-06	2.96e-06	2.96e-06	2.96e-06	2.96e-06
$b \bar{b} \rightarrow H H$	998	6.33e-06	6.33e-06	6.33e-06	6.33e-06	6.33e-06	6.33e-06	6.33e-06	6.33e-06	6.33e-06	6.33e-06

Notes: The results for CALCHEP, MADGRAPH/MADEVENT, WHIZARD 1.x and WHIZARD 2.0 are labeled by CH, MG, WO1 and WO2, respectively. ‘F’ and ‘U’ refer to the choice of the gauge in which the calculation was performed, Feynman or unitarity. The p_T cuts were fixed according to (2), and the unit for all cross sections is pb

Table 5 Cross sections for a selection of two-to-two processes in the MHM

Process	\sqrt{s} [GeV]	FEYNRULES							Stock
		MG	CH	WO2	WO1	WO1	CH	WO2	WO2
		U	U	U	F	U	F	F	U
$g g \rightarrow g g$	200	1.88e04	1.88e04	1.88e04	1.88e04	1.88e04	1.88e04	1.88e04	1.88e04
$Z W^+ \rightarrow Z' W^{+'}$	4693	103	103	103	103	103	103	103	103
$\gamma \gamma \rightarrow W^+ W^-$	643	16.1	16.1	16.1	16.1	16.1	16.1	16.1	16.1
$Z W^+ \rightarrow W^+ Z'$	3015	0.755	0.755	0.755	0.755	0.755	0.755	0.755	0.756
$\gamma Z \rightarrow W^{+'} W^{-'}$	8730	0.0412	0.0412	0.0411	0.0412	0.0412	0.0412	0.0412	0.0412
$\gamma Z \rightarrow W^+ W^{-'}$	2686	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
$W^+ W^{+'} \rightarrow W^{+'} W^{+'}$	6322	3.32	3.31	3.31	3.31	3.31	3.31	3.31	3.32
$Z Z' \rightarrow W^{+'} W^{-'}$	6371	3.86	3.86	3.86	3.86	3.86	3.86	3.86	3.86
$W^+ W^+ \rightarrow W^+ W^{+'}$	2965	2.41	2.42	2.42	2.42	2.42	2.42	2.42	2.42
$\gamma W^+ \rightarrow Z' W^{+'}$	8656	0.0349	0.0349	0.0349	0.0349	0.0349	0.0349	0.0349	0.0349
<hr/>									
$W^+ W^{-'} \rightarrow s \bar{s}'$	18765	33.6	33.6	33.6	33.6	33.6	33.6	33.6	33.6
$Z' Z' \rightarrow b \bar{b}$	8094	0.0958	0.0959	0.0959	0.0959	0.0959	0.0959	0.0959	0.0959
$Z' Z' \rightarrow e^{-'} e^{+'}$	36900	1.54	1.54	1.54	1.54	1.54	1.54	1.54	1.54
$\gamma W^{+'} \rightarrow c' \bar{s}'$	34886	0.00220	0.00220	0.00220	0.00220	0.00220	0.00220	0.00220	0.00220
$W^+ W^{-'} \rightarrow \tau^{-'} \tau^{+'}$	35208	1.45	1.45	1.45	1.45	1.45	1.45	1.45	1.45
$W^+ W^{-'} \rightarrow u \bar{u}'$	18765	34.0	34.0	34.0	34.0	34.0	34.0	34.0	34.0
$Z Z \rightarrow u \bar{u}'$	17173	30.3	30.3	30.3	30.3	30.3	30.3	30.3	30.3
$W^{+'} W^{-'} \rightarrow b \bar{b}'$	20460	38.5	38.5	38.5	38.5	38.5	38.5	38.5	38.4
$W^{+'} W^{-'} \rightarrow \nu_e' \bar{\nu}_e'$	36886	1.64	1.64	1.64	1.64	1.64	1.64	1.64	1.64
$\gamma \gamma \rightarrow \mu^{-'} \mu^{+'}$	32886	0.000463	0.000463	0.000463	0.000463	0.000463	0.000463	0.000463	0.000463
<hr/>									
$\nu_\tau u \rightarrow \nu_\tau' u'$	32886	0.0966	0.0966	0.0967	0.0966	0.0966	0.0966	0.0966	0.0966
$\nu_e u \rightarrow \nu_e' u'$	32886	0.0966	0.0966	0.0966	0.0966	0.0966	0.0966	0.0967	0.0967
$c \bar{c} \rightarrow \nu_e' \bar{\nu}_e'$	32896	5.38e-06	5.38e-06	5.38e-06	5.38e-06	5.38e-06	5.38e-06	5.38e-06	5.38e-06
$b u' \rightarrow t' d'$	49606	0.0337	0.0337	0.0337	0.0337	0.0337	0.0337	0.0337	0.0337
$\tau^- \bar{d} \rightarrow \nu_\tau' \bar{u}'$	32893	0.386	0.386	0.386	0.386	0.386	0.386	0.386	0.386
$e^- b \rightarrow e^{-'} b'$	32903	0.0965	0.0965	0.0965	0.0965	0.0965	0.0965	0.0966	0.0965
$t \bar{\nu}_\tau \rightarrow \tau^{+'} b'$	50014	0.0430	0.0429	0.0429	0.0429	0.0429	0.0429	0.0429	0.0429
$c \bar{s} \rightarrow \nu_e' e^{+'}$	32892	1.82e-05	1.82e-05	1.82e-05	1.82e-05	1.82e-05	1.82e-05	1.82e-05	1.82e-05
$\nu_\tau \bar{\nu}_\tau \rightarrow t \bar{t}'$	17388	7.23e-06	7.24e-06	7.24e-06	7.24e-06	7.24e-06	7.24e-06	7.24e-06	7.24e-06
$\tau^- \tau^+ \rightarrow s \bar{s}'$	16458	9.33e-06	9.35e-06	9.35e-06	9.35e-06	9.35e-06	9.35e-06	9.35e-06	9.35e-06

Notes: The results for CALCHEP, MADGRAPH/MADEVENT, WHIZARD 1.x and WHIZARD 2.0 are denoted by CH, MG, WO1 and WO2, respectively. ‘F’ and ‘U’ refer to the choice of the gauge in which the calculation was performed, Feynman or unitarity. The p_T cuts were fixed according to (2), and the unit for all cross sections is pb. A trailing prime “’” at the end of a particle identifier denotes its heavy partner (cf. Ref. [46])

WOGauge

Gauge choice (cf. Sect. 3.2). Possible values are:

WOFeynman, WORxi and WOUunitarity (default).

WOGaugeParameter

The external or internal parameter representing the gauge ξ in the R_ξ gauges (cf. Sect. 3.2). Default: Rx1

WOAutoGauge

Automatically assign the Goldstone boson masses in the Feynman and R_ξ gauges and automatically append the

symbol for ξ to the parameter list in the R_ξ gauges.

Default: False

WOMaxNcf

Maximum number of color flows provided for in the output. Essentially, this is $n_8 - \frac{1}{2}n_3$ where n_8 is the maximum number of external color octets and n_3 is the maximum number of external triplets and antitriplets. Note that this only affects WHIZARD 1.9x; there is no limit for 2.0.0+. Default: 4

Table 6 Input parameters for the SM

Parameter	Symbol	Value
Inverse of the electromagnetic coupling	$\alpha_{EW}^{-1}(M_Z)$	127.9
Strong coupling	$\alpha_s(M_Z)$	0.1172
Fermi constant	G_F	$1.16639 \times 10^{-5} \text{ GeV}^{-2}$
Z pole mass	M_Z	91.188 GeV
c quark mass	m_c	0
b quark mass	m_b	4.7 GeV
t quark mass	m_t	174.3 GeV
τ lepton mass	m_τ	1.777 GeV
Higgs mass	m_H	120 GeV
Cabibbo angle	θ_c	0.227736

Table 7 Input parameters for the MHM

Parameter	Symbol	Value
Inverse of the electromagnetic coupling	$\alpha_{EW}^{-1}(M_Z)$	127.9
Strong coupling	$\alpha_s(M_Z)$	0.1172
Z mass	M_Z	91.1876 GeV
W mass	M_W	80.398 GeV
W' mass	$M_{W'}$	500 GeV
Heavy fermion mass	M_F	4 TeV
μ lepton mass	m_μ	0.1057 GeV
τ lepton mass	m_τ	1.777 GeV
s quark mass	m_s	0.104 GeV
c quark mass	m_c	1.27 GeV
b quark mass	m_b	4.2 GeV
t quark mass	m_t	171.2 GeV

WORunParameters

The list of all internal parameters which will be recalculated if α_S is evolved (see above). This only affects WHIZARD 2.0.0+. Default: {a_S, G}

WOFast

If the interface drops vertices which are supported, this option can be set to False to enable some more time consuming checks which might aid the identification. Default: True

WOMaxCouplingsPerFile

The maximum number of couplings that are written to a single FORTRAN file. If compilation takes too long or fails, this can be lowered. Default: 500

WOverbose

Enable verbose output and in particular more extensive information on any skipped vertices. Default: False

Appendix B: Validation tables

In this appendix we summarize the validation of the interface from FEYNRULES to WHIZARD which was performed by calculating the integrated cross sections related to all possible two-to-two partonic processes in the SM, the MSSM and the Minimal Higgsless Model. We evaluated this quantity both with the FEYNRULES-generated model files and the built-in (“stock”) implementations of the various matrix-element generators, if existing, in the framework of CALCHEP, MADGRAPH/MADEVENT and WHIZARD/O’MEGA. Where supported by both model and generator, we also checked both Feynman and unitarity gauge. We ran the codes such that the Monte Carlo uncertainty was pushed below one percent, and excellent agreement at this level was achieved in all cases. In Tables 4, 5 and 8 we present a selection of the results we obtained. The full list of results is available from the FEYNRULES website.

Table 8 Cross sections for a selection of two-to-two processes in the MSSM

Process	\sqrt{s} [GeV]	FEYNRULES				Stock	
		CH	MG	WO1	WO2	MG	WO2
		U	U	U	U	U	U
$Z Z \rightarrow W^+ W^-$	1368	26.2	26.3	26.2	26.2	26.3	26.2
$W^+ W^+ \rightarrow W^+ W^+$	1277	25.6	25.7	25.6	25.7	25.7	25.6
$\gamma Z \rightarrow W^+ W^-$	1003	19.3	19.3	19.3	19.3	19.3	19.3
$g g \rightarrow g g$	800	839	840	839	839	839	839
$Z Z \rightarrow Z Z$	1459	0.223	0.223	0.223	0.223	0.223	0.223
$\gamma \gamma \rightarrow W^+ W^-$	1278	4.95	4.96	4.95	4.95	4.96	4.95
$\tilde{g} \tilde{g} \rightarrow g g$	4862	0.977	0.977	0.977	0.977	0.977	0.977
$\tau^- \tau^+ \rightarrow \gamma Z$	758	0.327	0.327	0.327	0.327	0.327	0.327
$\tilde{\chi}_4^0 \tilde{\chi}_1^+ \rightarrow \gamma W^+$	2573	0.0219	0.0219	0.0219	0.0219	0.0219	0.0219
$d \bar{d} \rightarrow Z Z$	730	0.0838	0.0839	0.0838	0.0838	0.0839	0.0839
$t \bar{t} \rightarrow g g$	1400	6.00	6.00	6.00	5.99	6.00	5.99
$c \bar{c} \rightarrow \gamma \gamma$	800	0.0272	0.0272	0.0272	0.0272	0.0272	0.0272
$b \bar{b} \rightarrow W^+ W^-$	678	1.72	1.72	1.72	1.72	1.72	1.72
$\tilde{\chi}_1^0 \tilde{\chi}_1^0 \rightarrow W^+ W^-$	1412	0.00586	0.00585	0.00586	0.00586	0.00587	0.00586
$c \bar{s} \rightarrow \tilde{\chi}_1^0 \tilde{\chi}_1^+$	1114	0.000873	0.000874	0.000873	0.000873	0.000874	0.000873
$\tilde{\chi}_2^0 \tilde{\chi}_2^0 \rightarrow \tilde{\chi}_2^0 \tilde{\chi}_2^0$	2897	0.00237	0.00237	0.00237	0.00237	0.00236	0.00237
$\tilde{\chi}_1^+ \tilde{\chi}_2^+ \rightarrow \tilde{\chi}_2^+ \tilde{\chi}_2^+$	5286	0.0151	0.0151	0.0151	0.0151	0.0151	0.0151
$c \bar{c} \rightarrow \tilde{\chi}_2^0 \tilde{\chi}_3^0$	2179	0.000134	0.000134	0.000134	0.000134	0.000134	0.000134
$s \bar{s} \rightarrow \tilde{g} \tilde{g}$	4862	0.144	0.144	0.144	0.144	0.144	0.144
$\mu^- \mu^+ \rightarrow \tilde{\chi}_2^0 \tilde{\chi}_2^0$	1449	0.0291	0.0291	0.0291	0.0291	0.0291	0.0291
$t \bar{b} \rightarrow \tilde{\chi}_3^0 \tilde{\chi}_2^+$	3694	0.00759	0.00759	0.00758	0.00759	0.00759	0.00759
$c \bar{c} \rightarrow \tilde{g} \tilde{g}$	4862	0.144	0.144	0.144	0.144	0.144	0.144
$\tau^- u \rightarrow \tilde{l}_6^- \tilde{u}_5$	3079	0.000510	0.000510	0.000510	0.000510	0.000510	0.000510
$\nu_\mu \nu_\tau \rightarrow \tilde{\nu}_1 \tilde{\nu}_2$	1480	0.00902	0.00903	0.00902	0.00902	0.00903	0.00902
$\gamma Z \rightarrow \tilde{l}_1^- \tilde{l}_6^+$	1730	0.00278	0.00278	0.00278	0.00278	0.00278	0.00278
$s \tilde{\chi}_2^0 \rightarrow \tilde{d}_3 H_1$	3349	9.85e-06	9.85e-06	9.85e-06	9.85e-06	9.85e-06	9.85e-06
$\gamma Z \rightarrow \tilde{u}_5 \tilde{u}_5^*$	4854	0.00268	0.00268	0.00268	0.00268	0.00268	0.00268
$\nu_e e^- \rightarrow \tilde{\nu}_3 \tilde{l}_2^-$	1317	0.0274	0.0274	0.0274	0.0274	0.0274	0.0274
$s \tilde{\chi}_1^0 \rightarrow \tilde{u}_4 H^-$	4263	9.84e-06	9.84e-06	9.84e-06	9.84e-06	9.84e-06	9.83e-06
$\tau^- \tau^+ \rightarrow \tilde{u}_1 \tilde{u}_6^*$	3956	0.000449	0.000449	0.000449	0.000449	0.000449	0.000449

Notes: The results for CALCHEP, MADGRAPH/MADEVENT, WHIZARD 1.x and WHIZARD 2.0 are denoted by CH, MG, WO1 and WO2, respectively. ‘F’ and ‘U’ refer to the choice of the gauge in which the calculation was performed, Feynman or unitarity. The p_T cuts were fixed according to (2), and the unit for all cross sections is pb. The sfermions are labeled according to the SLHA2 conventions (cf. Ref. [48])

In order to avoid collinear divergences, we imposed a cut on the p_T of each final state particle,

$$p_T \geq \frac{\sqrt{s}}{4}. \quad (2)$$

For the MSSM, the parameter set used corresponds to the benchmark point SPS1a [49] (with the strong coupling set to $\alpha_s(M_z) = 0.1172$), whereas for the SM and MHM the values of the input parameters are summarized in Tables 6 and 7.

The strong coupling was evolved to \sqrt{s} for each process. All particle widths were set to zero.

References

1. A. Pukhov et al., arXiv:[hep-ph/9908288](#)
2. E. Boos et al. (CompHEP Collaboration), Nucl. Instrum. Methods A **534**, 250 (2004). arXiv:[hep-ph/0403113](#)
3. A. Pukhov, arXiv:[hep-ph/0412191](#)

4. T. Stelzer, W.F. Long, *Comput. Phys. Commun.* **81**, 357 (1994). arXiv:[hep-ph/9401258](#)
5. F. Maltoni, T. Stelzer, *J. High Energy Phys.* **0302**, 027 (2003). arXiv:[hep-ph/0208156](#)
6. J. Alwall et al., *J. High Energy Phys.* **0709**, 028 (2007). arXiv:[0706.2334](#) [hep-ph]
7. J. Alwall, P. Artoisenet, S. de Visscher, C. Duhr, R. Frederix, M. Herquet, O. Mattelaer, *AIP Conf. Proc.* **1078**, 84 (2009). arXiv:[0809.2410](#) [hep-ph]
8. J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, T. Stelzer, *J. High Energy Phys.* **1106**, 128 (2011). arXiv:[1106.0522](#) [hep-ph]
9. T. Gleisberg, S. Hoche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, J. Winter, *J. High Energy Phys.* **0902**, 007 (2009). arXiv:[0811.4622](#) [hep-ph]
10. M. Moretti, T. Ohl, J. Reuter, arXiv:[hep-ph/0102195](#)
11. W. Kilian, T. Ohl, J. Reuter, *Eur. Phys. J. C* **71**, 1742 (2011). arXiv:[0708.4233](#) [hep-ph]
12. N.D. Christensen, C. Duhr, *Comput. Phys. Commun.* **180**, 1614 (2009). arXiv:[0806.4194](#) [hep-ph]
13. N.D. Christensen, P. de Aquino, C. Degrande, C. Duhr, B. Fuks, M. Herquet, F. Maltoni, S. Schumann, *Eur. Phys. J. C* **71**, 1541 (2011). arXiv:[0906.2474](#) [hep-ph]
14. J.M. Butterworth et al. arXiv:[1003.1643](#) [hep-ph]
15. C. Duhr, B. Fuks, *Comput. Phys. Commun.* **182**, 2404 (2011). arXiv:[1102.4191](#) [hep-ph]
16. G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, arXiv:[1111.2034](#) [hep-ph]
17. C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer, T. Reiter, arXiv:[1108.2040](#) [hep-ph]
18. W. Kilian, J. Reuter, S. Schmidt, D. Wiesler, *J. High Energy Phys.* **1204**, 013 (2012)
19. N. Greiner, A. Guffanti, T. Reiter, J. Reuter, *Phys. Rev. Lett.* **107**, 102002 (2011). arXiv:[1105.3624](#) [hep-ph]
20. J.R. Andersen et al. (SM and NLO Multileg Working Group), arXiv:[1003.1241](#) [hep-ph]
21. T. Binoth, N. Greiner, A. Guffanti, J. Reuter, J.P. Guillet, T. Reiter, *Phys. Lett. B* **685**, 293 (2010). arXiv:[0910.4379](#) [hep-ph]
22. T. Ohl, C. Speckner, *Phys. Rev. D* **82**, 116011 (2010). arXiv:[1008.4710](#) [hep-ph]
23. J. Reuter, F. Braam, *AIP Conf. Proc.* **1200**, 470 (2010). arXiv:[0909.3059](#) [hep-ph]
24. T. Ohl, C. Speckner, *Phys. Rev. D* **78**, 095008 (2008). arXiv:[0809.0023](#) [hep-ph]
25. J. Kalinowski, W. Kilian, J. Reuter, T. Robens, K. Rolbiecki, *J. High Energy Phys.* **0810**, 090 (2008). arXiv:[0809.3997](#) [hep-ph]
26. T. Robens, J. Kalinowski, K. Rolbiecki, W. Kilian, J. Reuter, *Acta Phys. Pol. B* **39**, 1705 (2008). arXiv:[0803.4161](#) [hep-ph]
27. A. Alboteanu, W. Kilian, J. Reuter, *J. High Energy Phys.* **0811**, 010 (2008). arXiv:[0806.4145](#) [hep-ph]
28. W. Kilian, D. Rainwater, J. Reuter, *Phys. Rev. D* **74**, 095003 (2006). arXiv:[hep-ph/0609119](#)
29. W. Kilian, J. Reuter, T. Robens, *Eur. Phys. J. C* **48**, 389 (2006). arXiv:[hep-ph/0607127](#)
30. M. Beyer, W. Kilian, P. Krstonsic, K. Monig, J. Reuter, E. Schmidt, H. Schroder, *Eur. Phys. J. C* **48**, 353 (2006). arXiv:[hep-ph/0604048](#)
31. W. Kilian, D. Rainwater, J. Reuter, *Phys. Rev. D* **71**, 015008 (2005). arXiv:[hep-ph/0411213](#)
32. T. Ohl, J. Reuter, *Phys. Rev. D* **70**, 076007 (2004). arXiv:[hep-ph/0406098](#)
33. K. Hagiwara et al., *Phys. Rev. D* **73**, 055005 (2006). arXiv:[hep-ph/0512260](#)
34. J.A. Aguilar-Saavedra et al., *Eur. Phys. J. C* **46**, 43 (2006). arXiv:[hep-ph/0511344](#)
35. T. Ohl, J. Reuter, *Eur. Phys. J. C* **30**, 525 (2003). arXiv:[hep-th/0212224](#)
36. M.R. Whalley, D. Bourilkov, and R. C. Group, arXiv:[hep-ph/0508110](#)
37. T. Sjostrand, S. Mrenna, P.Z. Skands, *J. High Energy Phys.* **0605**, 026 (2006). arXiv:[hep-ph/0603175](#)
38. M. Dobbs, J.B. Hansen, *Comput. Phys. Commun.* **134**, 41 (2001)
39. H. Murayama, I. Watanabe, K. Hagiwara, KEK-91-11
40. F. Caravaglios, M. Moretti, *Z. Phys. C* **74**, 291 (1997). arXiv:[hep-ph/9604316](#)
41. F. Maltoni, K. Paul, T. Stelzer, S. Willenbrock, *Phys. Rev. D* **67**, 014026 (2003). arXiv:[hep-ph/0209271](#)
42. W. Kilian, T. Ohl, J. Reuter, C. Speckner, in preparation
43. T. Ohl, *Comput. Phys. Commun.* **120**, 13 (1999). arXiv:[hep-ph/9806432](#)
44. G.P. Lepage, CLNS-80/447
45. T. Han, J.D. Lykken, R.J. Zhang, *Phys. Rev. D* **59**, 105006 (1999). arXiv:[hep-ph/9811350](#)
46. R.S. Chivukula, B. Coleppa, S. Di Chiara, E.H. Simmons, H.J. He, M. Kurachi, M. Tanabashi, *Phys. Rev. D* **74**, 075011 (2006). arXiv:[hep-ph/0607124](#)
47. P.Z. Skands et al., *J. High Energy Phys.* **0407**, 036 (2004). arXiv:[hep-ph/0311123](#)
48. B.C. Allanach et al., *Comput. Phys. Commun.* **180**, 8 (2009). arXiv:[0801.0045](#) [hep-ph]
49. B.C. Allanach et al., *Eur. Phys. J. C* **25**, 113 (2002). arXiv:[hep-ph/0202233](#). Proc. of the APS/DPF/DPB Summer Study on the Future of Particle Physics (Snowmass 2001), ed. by N. Graf
50. B. Fuks, *Int. J. Mod. Phys. A* **27**, 1230007 (2012). arXiv:[1202.4769](#) [hep-ph]
51. R.S. Chivukula, E.H. Simmons, H.-J. He, M. Kurachi, M. Tanabashi, *Phys. Rev. D* **72**, 015008 (2005). arXiv:[hep-ph/0504114](#)
52. H.-J. He, Y.-P. Kuang, Y.-H. Qi, B. Zhang, A. Belyaev, R.S. Chivukula, N.D. Christensen, A. Pukhov et al., *Phys. Rev. D* **78**, 031701 (2008). arXiv:[0708.2588](#) [hep-ph]
53. C. Speckner, arXiv:[1011.1851](#) [hep-ph]
54. N. Christensen, B. Fuks, J. Reuter, C. Speckner, arXiv:[1204.6264](#) [hep-ph]