**THE EUROPEAN
PHYSICAL JOURNAL C**

Special Article - Tools for Experiment and Theory

Automated mass spectrum generation for new physics

Adam Alloul[1], **Jorgen D'Hondt**[2], **Karen De Causmaecker**[2], **Benjamin Fuks**[1,3,a], **Michel Rausch de Traubenberg**[1]

[1]Institut Pluridisciplinaire Hubert Curien/Département Recherches Subatomiques, Université de Strasbourg/CNRS-IN2P3, 23 Rue du Loess, 67037 Strasbourg, France
[2]Interuniversity Institute for High Energies (IIHE), ELEM, and Theoretische Natuurkunde, Vrije Universiteit Brussel and The International Solvay Institutes, Pleinlaan 2, 1050 Brussels, Belgium
[3]Theory Division, Physics Department, CERN, 1211 Geneva 23, Switzerland

Abstract We describe an extension of the FEYNRULES package dedicated to the automatic generation of the mass spectrum associated with any Lagrangian-based quantum field theory. After introducing a simplified way to implement particle mixings, we present a new class of FEYNRULES functions allowing both for the analytical computation of all the model mass matrices and for the generation of a C++ package, dubbed ASPERGE. This program can then be further employed for a numerical evaluation of the rotation matrices necessary to diagonalize the field basis. We illustrate these features in the context of the Two-Higgs-Doublet Model, the Minimal Left-Right Symmetric Standard Model and the Minimal Supersymmetric Standard Model.

1 Introduction

Although the Standard Model of particle physics is very well verified empirically at the current accessible energies, numerous extensions to its Lagrangian are proposed. These extensions describe new or alternative fundamental interactions that typically accommodate possible new physics phenomena at higher energies as well as at the current collider energies. In this top-down approach, the phenomenology of the proposed extensions is to be confronted with experimental observations. In order to obtain the mass spectrum of any new physics model reflected by its Lagrangian, the mixing matrices of the gauge eigenstates into the mass eigenstates are needed. An automated mass spectrum generator, AS-PERGE[1] is developed within the framework of the FEYN-RULES program [1–5] to determine the mixing matrices nu-

merically. This allows for a study of the direct relation between the parameters of any new physics model and the observable masses of the fundamental particles.

This paper describes in Sects. 2 and 3 the relevant parts of the FEYNRULES program to introduce the new ASPERGE package, detailed in Sect. 4. To illustrate its application several examples are presented in Sect. 5.

2 The FEYNRULES package

The program FEYNRULES [1–5] is a MATHEMATICA[2] package that allows for the automated extraction of Feynman rules from any Lagrangian describing the dynamics of a perturbative quantum field theory. The Feynman rules, together with general information such as the definitions of the model particles or of the Lagrangian parameters, can subsequently be exported by means of several translation interfaces to matrix-element generators. Up to now, interfaces to COMPHEP and CALCHEP [6–9], FEYNARTS and FORMCALC [10–13], MADGRAPH and MADEVENT [14–18], SHERPA [19, 20] and WHIZARD [21, 22] have been developed.

In addition, any model can also be converted to a PYTHON library containing classes and objects representing particles, parameters and vertices. This format is dubbed the Universal FEYNRULES Output (UFO) format [23] and is appropriate to address the implementation of any high-energy physics model into computational tools. Its strength lies in its agnosticism with respect to the allowed Lorentz and/or color structures appearing in the Lagrangian, in contrast to any other more conventional model format for which restrictions are imposed. Presently, the UFO is used by ALOHA [24], MADANALYSIS 5 [25], and MADGRAPH 5 [18], and

---

[1]The acronym ASPERGE stands for **A**utomated **Spe**ct**r**um **Ge**neration.

a e-mail: benjamin.fuks@iphc.cnrs.fr

[2]MATHEMATICA is a registered trademark of Wolfram Research, Inc.

will be used, in the future, by GOSAM [26, 27] and HER-WIG++ [28].

The FEYNRULES model structure extends the format employed in FEYNARTS [11] so that particles, parameters and gauge groups are now all defined in a similar fashion. Following the FEYNARTS conventions, particles are collected into classes describing multiplets having exactly the same quantum numbers, but possibly different masses. Each of these classes is defined as a set of MATHEMATICA replacement rules referring to its properties. For example, the three vector fields W_i associated to the $SU(2)_L$ gauge subgroup of the Standard Model could be declared as

```
V[1] == {
  ClassName     -> Wi,
  Unphysical    -> True,
  SelfConjugate -> True,
  Indices       -> { Index[SU2W] },
  FlavorIndex   -> SU2W,
  Definitions   -> {
   Wi[mu_,1]->(Wbar[mu]+W[mu])/Sqrt[2],
   Wi[mu_,2]->(Wbar[mu]-W[mu])/(I*Sqrt[2]),
   Wi[mu_,3]->cw Z[mu] + sw A[mu]}
}
```

This set of MATHEMATICA rules defines a vector field (V[1]) represented by the symbol Wi (its ClassName) and carrying a flavor index SU2W associated with the adjoint gauge index of $SU(2)_L$. As this field is declared as unphysical[3] (Unphysical->True), it must be linked to one or several of the mass eigenstates of the model by means of appropriate mixing relations. As illustrated in the example, these relations are passed through the attribute Definitions of the particle class. They can either be purely numerical, as for the W_1 and W_2 bosons that are rotated to the charged W^+ and W^- bosons, or depend on some of the model parameters, as for the W_3 field which is re-expressed in terms of the photon A and the Z boson through a relation depending on the sine and cosine of the electroweak mixing angle (sw and cw).

In contrast, declaring physical particles requires the implementation of extra information such as their masses (Mass), widths (Width) and Particle Data Group codes (PDG) [29]. The Z-boson field introduced above could hence be declared as

```
V[2] == {
  ClassName     -> Z,
  SelfConjugate -> True,
  Mass          -> {MZ, 91.1876},
  Width         -> {WZ, 2.4952},
  PDG           -> 23
}
```

---

[3]In this work, we denote as unphysical any field that is not a mass eigenstate of the theory.

The declaration of the model parameters and gauge group is similar and based on dedicated classes with their own set of attributes. Since only the particle class properties introduced above are sufficient for the understanding of the present work, we omit any further detail and refer the reader to Refs. [1, 4] for more information on particle, parameter and gauge group implementation in FEYNRULES.

The last key ingredient to achieve a model implementation consists of its Lagrangian. It is provided using standard MATHEMATICA commands, augmented by some special symbols representing objects such as Dirac matrices, vector field strength tensors or covariant derivatives. The user has then the possibility to perform basic checks on the Lagrangian, such as verifying its hermiticity, the normalization of kinetic terms, *etc.* We again refer to the FEYNRULES manual for more information [1].

After having imported the FEYNRULES package into the current MATHEMATICA session, the model is loaded by issuing

```
LoadModel[ "file1.fr", "file2.fr", ... ]
```

where its implementation can be possibly spread among the files "file1.fr", "file2.fr", *etc.*, according to the convenience of the user. It can also be directly implemented within the MATHEMATICA session so that the function LoadModel is called, in that case, without any argument.

The Feynman rules can be subsequently extracted by means of the command

```
FeynmanRules[ Lag ]
```

where Lag is the MATHEMATICA symbol containing the expression of the Lagrangian, written in four-dimensional spacetime and employing four-component spinors for fermions. The FeynmanRules method extracts the interaction vertices included in the Lagrangian Lag so that they can be further employed within MATHEMATICA for dedicated studies.

All the interfaces to Monte Carlo event generation tools can be invoked in a similar procedure,

```
WriteXXX[ Lag ]
```

where the sequence of letters XXX takes one of the values CHOutput (CALCHEP), FeynArtsOutput (FEYN-ARTS), SHOutput (SHERPA), UFO or WOOutput (WHIZARD).

In the context of supersymmetric theories, the most natural and convenient way to construct a Lagrangian consists of employing the superspace formalism. Therefore, the FEYN-RULES package includes a module allowing for superfield declarations and Lagrangian implementation in terms of superfields. Dedicated functions are then provided to convert superfield expressions into a form more suitable with respect to the requirements of the interfaces to the Monte Carlo event generators [4].

3 Implementing mixings in FEYNRULES

In this work, we propose an extension of the FEYNRULES package aiming to simplify the declaration of the mixing relations linking the unphysical degrees of freedom of the theory to the physical fields. This new module allows to automatically fill the Definitions attribute of the fields, where relevant, and declare the mixing matrices as parameters. In addition, we have developed an interface generating a C++ code dedicated to the diagonalization of the mass matrices of the model (see Sect. 4) after having implemented in FEYNRULES a function allowing for their analytical extraction from the Lagrangian. In this way, the values of all the mixing parameters are derived numerically and can be reimported into FEYNRULES.

3.1 Mixing declarations

For an efficient declaration of the mixing relations, we have extended the FEYNRULES model file structure by adjoining a new class dedicated to particle mixings. Consequently, all mixing relations among the states can be declared on the same spirit as particles, gauge groups and parameters, after having been gathered into a list dubbed M$MixingsDescription

```
M$MixingsDescription = {
   Mix["l1"] == { options1 },
   Mix["l2"] == { options2 },
        ...
}
```

Each element of this list consists of an equality dedicated to one specific mixing relation. It associates a label, given as a string, ("l1", "l2", *etc.*) with a set of MATHEMATICA replacement rules defining the mixing properties (options1, options2, *etc.*).

In order to illustrate the choice of options offered to the user, we consider the example of Sect. 2 where we have focused on the mixing of the $SU(2)_L$ gauge bosons. We start by implementing the mixing of the W_1 and W_2 gauge fields,

$$W_\mu^+ = \frac{W_\mu^1 - i W_\mu^2}{\sqrt{2}} \quad \text{and} \quad W_\mu^- = \frac{W_\mu^1 + i W_\mu^2}{\sqrt{2}}, \quad (1)$$

which stems from the diagonalization of the third generator of $SU(2)_L$ in the adjoint representation. As Eq. (1) is purely numerical, i.e., it does not involve any model parameter, it can be declared in the model file in a very compact form,

```
Mix["Wmix"] == {
  MassBasis  -> {W, Wbar},
  GaugeBasis -> {Wi[1], Wi[2]},
  Value      -> { {1/Sqrt[2], -I/Sqrt[2]},
                  {1/Sqrt[2], I/Sqrt[2]}  }
}
```

The command above declares a mixing relation, dubbed Wmix, that can be schematically written as

```
MassBasis = Value . GaugeBasis
```

where the dot product stands for the usual matrix product. The information on the gauge basis is provided as the value of the attribute GaugeBasis which refers here to the unphysical fields W_1 (Wi[1]) and W_2 (Wi[2]). Similarly, the MassBasis attribute refers to the mass basis, containing here the symbols associated with the W^+ (W) and W^- (Wbar) bosons. Finally, the mixing matrix is given under a numerical form as the argument of the attribute Value.

Some remarks are in order. First, the gauge basis only contains unphysical fields, while the mass basis can contain either physical fields, unphysical fields or both. Particle mixings can therefore be possibly implemented in several steps, as illustrated in Sect. 5.2. Next, spin and Lorentz indices can be omitted and the index ordering is defined when declaring the fields (through the attribute Indices of the particle class [1]). Finally, if some indices are irrelevant, i.e., if they are identical for all the involved fields, underscores can be employed to simplify the mixing declaration. For instance, the three left-handed down-type squarks $\tilde{d}_L^c$ (sdL[1,c]), $\tilde{s}_L^c$ (sdL[2,c]), and $\tilde{b}_L^c$ (sdL[3,c]) are related to the squark gauge-eigenstates $\tilde{Q}_L^{ifc}$ (QLs[i,f,c]), the index i being a fundamental $SU(2)_L$ index, the index f a flavor index and the index c a fundamental color index. The corresponding declaration reads

```
Mix["sdleft"] == {
  MassBasis  ->
    {sdL[1,_], sdL[2,_], sdL[3,_]},
  GaugeBasis ->
    {QLs[2,1,_], QLs[2,2,_], QLs[2,3,_]},
        ...
}
```

where the mixing matrix is the identity. The underscore reflects that the same color index is carried by all fields.

We now get back to weak gauge boson mixings and turn to the neutral sector. We hence focus on the rotation of the third weak boson W_3 and the hypercharge gauge boson B to the photon and Z-boson states,

$$\begin{pmatrix} A_\mu \\ Z_\mu \end{pmatrix} = U_w \begin{pmatrix} B_\mu \\ W_\mu^3 \end{pmatrix}, \quad (2)$$

after introducing the *a priori* unknown weak mixing matrix U_w[4]. The computation of the numerical values of its matrix

---

elements is addressed by means of the C++ package generated by FEYNRULES (see Sect. 4) and is only possible if the mixing is declared according to the syntax

```
Mix["AZmix"] == {
  MassBasis    -> {A, Z},
  GaugeBasis   -> {B, Wi[3]},
  MixingMatrix -> UW,
  BlockName    -> WEAKMIX
}
```

The declaration of the gauge and mass bases is similar to the case of the charged W bosons, while the attribute `Value` has been removed as the numerical value of the mixing matrix is not known. The user provides instead the symbol referring to the mixing matrix (`UW`) by means of the `MixingMatrix` attribute, *without* declaring it as one of the model parameters. This last task is internally handled by FEYNRULES which assumes that the mixing matrix is complex and which creates two external tensorial parameters, one for the real part and one for the imaginary part of the matrix, together with one internal tensorial parameter being the matrix itself.

When a symbol for a mixing matrix is provided, it is mandatory to specify, in addition, the name of a Les Houches block which will contain the numerical values associated with the elements of the matrix. We indeed recall that both FEYNRULES and most of the interfaced Monte Carlo event generators order the model parameters according to a structure inspired by the Supersymmetry Les Houches Accord (SLHA) [30, 31]. In our example, we impose the real part of the elements of U_w to be stored in a Les Houches block WEAKMIX and their imaginary part in an automatically created block IMWEAKMIX, i.e., a block of the same name with the prefix IM appended.

Implementing model Lagrangians might require to explicitly use one or several of the mixing matrices for some of the model interactions, as for the Minimal Supersymmetric Standard Model where the CKM matrix is employed in the superpotential [4]. In this case, the matrices must be declared according to the standard syntax presented in the FEYNRULES manual, numerical values being provided as inputs. This subsequently renders the attribute `BlockName` of the mixing class obsolete and ignored by FEYNRULES. Contrary, mixing matrices automatically declared through a mixing declaration cannot be employed in Lagrangians.

In the Standard Model, the CKM matrix V_{CKM} relates the left-handed down quark gauge-eigenstates d_L^0 to the mass-eigenstates d_L as

$$d_L^0 = V_{\text{CKM}} \cdot d_L. \tag{3}$$

To be compliant with the syntax presented so far, a symbol for the hermitian-conjugate matrix has to be created. To avoid such a complication, the optional attribute `Inverse`

can be used and set to `True`, which enforces a relation among the mass and gauge bases given by

```
GaugeBasis = MixingMatrix . MassBasis
```

3.2 More advanced cases

3.2.1 Scalar/pseudoscalar splittings

When neutral scalar fields are mixing, the gauge eigenstates in general split into their real degrees of freedom so that one scalar and one pseudoscalar mass basis are required. Consequently, a list of two bases is provided as argument of the `MassBasis` attribute, instead of a single basis as in Sect. 3.1. Consistently, the arguments of the attributes `Value`, `BlockName`, `MixingMatrix` and `Inverse` are also upgraded to lists. The first element of those lists always refers to the scalar fields, while the second one is related to the pseudoscalar fields. It may appear that some of the elements of those lists are irrelevant, as for instance when the scalar mixing matrix is unknown (`MixingMatrix` and `BlockName` are used) and the pseudoscalar mixing matrix is known (`Value` is used). The irrelevant list components are in this case replaced by underscores, as illustrated with

```
Mix["scalar"] == {
  MassBasis    -> { {h1, h2}, {a1, a2} },
  GaugeBasis   -> { phi1, phi2 },
  BlockName    -> { SMIX, _ },
  MixingMatrix -> { US, _ },
  Value        -> { _, ... }
}
```

where the (pseudo)scalar mass-eigenstates are represented by the symbols `h1` and `h2` (`a1` and `a2`) and the gauge eigenstates by `phi1` and `phi2`. In this example, the mixing matrix related to the scalar sector is denoted by `US` and is associated with the Les Houches block `SMIX`. Concerning the pseudoscalar sector, a numerical mixing matrix is instead provided (in the ellipses). A concrete example is given for the Two-Higgs-Doublet Model in Sect. 5.1.

3.2.2 Dirac and Weyl fermion mixings

Several options are left to the user concerning the implementation of Dirac fermion mixings. Either one single gauge basis is employed, so that FEYNRULES internally takes care of the chirality projectors that appear in the related mass terms, or different particle classes can be used for the left-handed and right-handed components of the fermions. In this case, the `GaugeBasis` attribute refers to a list of two gauge bases instead of to a single basis. For both options, the arguments of the attributes `Value`, `BlockName`, `Mixing-Matrix` and `Inverse` consist of lists, the first component being related to the mixing of the left-handed fermions and

the second one to the mixing of the right-handed fermions. As for neutral scalar mixing, underscores are used for irrelevant list elements.

Lagrangian mass terms for charged Weyl fermions are generically written as

$$
\left(\psi_1^-, \ldots, \psi_n^- \right) M \begin{pmatrix} \chi_1^+ \\ \vdots \\ \chi_n^+ \end{pmatrix}, \tag{4}
$$

where M stands for the mass matrix and ψ_i and χ_i are Weyl fermions which have been assigned an electric charge of ± 1 for the sake of the example. The diagonalization of the matrix M proceeds through two unitary rotations U and V,

$$
\begin{pmatrix} \tilde{\psi}_1^- \\ \vdots \\ \tilde{\psi}_n^- \end{pmatrix} = U \begin{pmatrix} \psi_1^- \\ \vdots \\ \psi_n^- \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \tilde{\chi}_1^+ \\ \vdots \\ \tilde{\chi}_n^+ \end{pmatrix} = V \begin{pmatrix} \chi_1^+ \\ \vdots \\ \chi_n^+ \end{pmatrix}, \tag{5}
$$

which introduces two mass bases. Therefore, all the attributes `MassBasis`, `GaugeBasis`, `Value`, `Mixing-Matrix`, and `BlockName` now take lists as arguments (with underscores included where relevant). The only extra rule to obey to is that the first components of these lists are associated with one of the two rotations and the second components with the second of them. An example is provided in Sect. 5.3.

3.3 Vacuum expectation value declarations

In realistic new physics models, the ground state of the theory is non-trivial and fields must be shifted by their vacuum expectation value. Since Lorentz invariance and electric charge conservation impose that only electrically neutral scalar fields can get non-vanishing vacuum expectation values, only shifts of (electrically) neutral scalar fields are allowed to be included in the mixing relations. This information is encompassed within the variable `M$vevs` which consists of a list of two-component elements. The first one refers to an unphysical field while the second one is the associated vacuum expectation value. For instance, the declaration of a configuration where two fields `phi1` and `phi2` get non-vanishing vacuum expectation values `vev1` and `vev2` could be performed as

```
M$vevs = { { phi1, vev1 }, { phi2, vev2 } }
```

The vacuum expectation values `vev1` and `vev2` must be declared as any other model parameter, as described in the FEYNRULES manual [1].

3.4 User functions

Once both the mixing relations and the vacuum expectation values have been properly declared, the mass matrices of the model can be extracted by means of the function `ComputeMassMatrix`,

```
ComputeMassMatrix[ Lag, options ]
```

where `Lag` is the model Lagrangian and the symbol `options` stands for optional arguments. If no option is provided, the function calculates all the mass matrices of the model for which the numerical value of the mixing matrix is unknown. It is possible to focus on a specific mixing relation whose label is denoted by `"l1"` by issuing, in MATHEMATICA,

```
ComputeMassMatrix[ Lag, Mix->"l1" ]
```

For the computation of multiple matrices, the label `"l1"` has to be replaced by a list of labels. During the computation of the mass matrices, a lot of information is by default printed to the screen. This can be avoided by including the optional argument `ScreenOutput -> False` in the two command lines above.

The input information and the result of the `Compute-MassMatrix` function can be retrieved through the intuitive printing functions, `MassMatrix`, `GaugeBasis`, `MassBasis`, `MixMatrix`, `BlockName` and `Matrix-Symbol` which all take as argument the label of a mixing relation. A wrapper is also available,

```
MixingSummary [ "l1" ]
```

which sequentially calls all the printing functions for a mixing relation represented by the label `"l1"` and organizes the output in a readable form.

The FEYNRULES method to extract analytically a mass matrix is fully generic and can be employed to compute any matrix M defined by the Lagrangian

$$
\mathcal{L}_{\mathrm{mass}} = \mathcal{B}_2^\dagger M \mathcal{B}_1, \tag{6}
$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ stand for two field bases possibly different. The calculation of the matrix M is achieved by issuing

```
ComputeMassMatrix[ Lag,
Basis1 -> b1, Basis2 -> b2 ]
```

where the symbols `b1` and `b2` are associated with the bases $\mathcal{B}_1$ and $\mathcal{B}_2$ and refer to lists of fields. In this case, the printing functions introduced above are not available.

4 Automated spectrum generation

4.1 The ASPERGE package

The computation of the unknown mixing matrices necessary for diagonalizing all the model mass matrices can in general only be achieved numerically. To this end, we have developed the C++ program ASPERGE. It includes a set of C++ source files (stored in the subdirectory `src`), coming together with the related header files (stored in the subdirectory `inc`), that can be split into model-independent and

model-dependent files. For an efficient use of the ASPERGE program, it has been entirely embedded within the FEYN-RULES package. Therefore, only a brief discussion of the structure of the code is presented in this paper. More information, such as a DOXYGEN documentation, can be found on the ASPERGE webpage [32].

The set of model-independent files contains, on the one hand, several tools dedicated to matrices and their diagonalization (`MassMatrix.cpp`, `MassMatrix.hpp` as well as `Matrix.hpp`). On the other hand, the ASPERGE code is based on an internal format for parameters, defined in the source files `Par.cpp`, `CPar.cpp`, `RPar.cpp` and in the associated header files. This format is inspired from a SLHA structure and the corresponding mapping is encoded into the files `ParSLHA.cpp`, `SLHABlock.cpp`, and in the associated header files. Finally, printing and string manipulation routines are included in the files `tools.cpp` and `tools.hpp` and the program comes with a makefile.

All the model dependency is included in the two files `Parameters.cpp` and `Parameters.hpp` as well as in the core program implemented in the `main.cpp` file.

The information encompassed in the two parameter files is threefold. First, the SLHA structure ordering the external parameters is encoded in terms of blocks and counters. Next, the definitions of the internal parameters as functions of the other model parameters are implemented, where a proper running of the ASPERGE program is only guaranteed if the parameters do not depend on the masses and mixing matrices to be computed. Finally, the analytical formulas of the mass matrices to diagonalize are included.

The main program (`main.cpp`) starts with the declaration of the different mass matrices of the model. Links to the relevant elements of the mass basis are then implemented by means of the associated PDG codes, which allow to assign the mass eigenvalues to each of the physical particles, the ordering of the PDG codes following the mass ordering.

4.2 Interfacing the ASPERGE package to FEYNRULES

The ASPERGE package can be entirely generated, for a given particle physics model, from the FEYNRULES model information by means of a dedicated interface which works as for the other FEYNRULES interfaces. It is then called by typing, in a MATHEMATICA session,

```
WriteASperGe[ Lag, Output -> dirname ]
```

where the symbol `Lag` stands for the model Lagrangian and `Output->dirname` for an optional argument indicating the name of the directory where to store all the created files. If unspecified, the directory `ModelName_MD` is employed, `ModelName` being the name of the FEYNRULES model.

The interface first extracts all the relevant mass matrices from the Lagrangian `Lag` by means of the function

`ComputeMassMatrix` introduced in Sect. 3.4. It then writes, in addition to model-independent files described in Sect. 4.1, the three model-dependent files `main.cpp`, `Parameters.cpp` and `Parameters.hpp`, together with one data file `Externals.dat` (stored in the subdirectory `input`). This last file contains the numerical values of the external parameters of the model, necessary for the numerical evaluation of the mass matrices. When running the code (see Sect. 4.3), the user can update this file or even employ a different file according to his needs.

The numerical matrix diagonalization performed by AS-PERGE is based on GSL functions relying on the hermiticity of the mass matrices which employs symmetric bi-diagonalization followed by QR reduction. This contrasts with existing diagonalization packages developed in the framework of FEYNARTS [33] and CALCHEP [34] that are based on Jacobi-type iterative algorithms. A hermiticity check is therefore performed by the interface before writing down the output. Since the mass matrix M related to charged fermions is by construction non-hermitian, the matrices $M^{\dagger}M$ and $MM^{\dagger}$ are employed instead, which allows to obtain left-handed and right-handed fermion mixing matrices separately.

It is also possible to focus on one or several specific mixing relations. In this case, the `Mix` option, already introduced in the context of the `ComputeMassMatrix` function, has to be used,

```
WriteASperGe[ Lag, Mix -> {"l1", "l2"} ]
```

We refer to Sect. 3.4 for more information.

4.3 Running ASPERGE

Since the ASPERGE package is based on GSL functions, it is mandatory to have the GSL libraries installed on the system. Then, if the `g++` compiler is available, the makefile generated by FEYNRULES can be employed directly. Otherwise, it must be first edited accordingly to include proper compiler information.

Once compiled, ASPERGE can be executed by typing in a shell

```
./ASperGe <infile> <outfile>
```

where the arguments indicate in which file the numerical value of the external parameters must be read (`<infile>`) and where to store the output file (`<outfile>`). This file contains, in addition to the input parameters, the computed numerical values of the mixing matrices, split in terms of their real and imaginary parts according to the SLHA conventions, as well as all the masses of the physical states (stored in the SLHA block `MASS`). In order to execute AS-PERGE with all the default settings as generated by the FEYNRULES interface, it is sufficient to type in a shell

```
./ASperGe input/externals.dat output/out.dat
```

Both the compilation and the execution of the program can be performed from the MATHEMATICA session, by issuing

```
RunASperGe[ ]
```

This also loads the SLHA parameter file `out.dat` back into the FEYNRULES session, so that it can be further employed, *e.g.*, to generate a UFO model. Information about the run of ASPERGE can be found in the file `ASperGe.log` stored in the same folder as the executable.

It is also possible to diagonalize specific mass matrices of the model by executing

```
./ASperGe <infile> <outfile> m1 m2 ...
```

where `m1`, `m2`, *etc.*, are the names of the mixing matrices to be computed.

5 Illustrative examples

In this section, we illustrate the features of the ASPERGE program and its interface to FEYNRULES by choosing three extensions of the Standard Model with non-trivial mixing relations, i.e., the Two-Higgs-Doublet Model (2HDM), the Minimal Left-Right Symmetric Standard Model (LRSM) and the Minimal Supersymmetric Standard Model (MSSM). We modify their original FEYNRULES implementations[5] [2, 4] to accommodate for the mixings as described in Sect. 3. We then employ the ASPERGE program (see Sect. 4) to numerically calculate some of the mass and mixing matrices of these models.

5.1 The general two-Higgs-doublet model

The 2HDM is one of the simplest extensions of the Standard Model, with respect to which it only contains a second weak doublet of scalar fields. Following the conventions of the original FEYNRULES implementation [2], both Higgs fields ϕ_1 and ϕ_2 carry the same hypercharge so that they can always be redefined by means of $U(2)$ transformations [35–38]. Adopting the so-called Higgs-basis, the two doublets read

$$\phi_1 = \begin{pmatrix} G^+ \\ \frac{v+H^0+iG^0}{\sqrt{2}} \end{pmatrix} \quad \text{and} \quad \phi_2 = \begin{pmatrix} H^+ \\ \frac{R^0+iI^0}{\sqrt{2}} \end{pmatrix}, \tag{7}$$

where only the neutral component of the ϕ_1 field acquires a vacuum expectation value v. Moreover, the Goldstone bosons $G^\pm$ and G^0 as well as the charged Higgs field $H^\pm$

---

[5]Since no previous implementation of the LRSM exists, we take the opportunity to provide the relevant details in Sect. 5.2.

are not required to be further rotated, so that only the mass matrix of the neutral fields H^0, R^0 and I^0 must still be diagonalized.

We extend the 2HDM FEYNRULES implementation described in Ref. [2] by first indicating that the second component of the ϕ_1 field, represented by the symbol `phi1`, acquires a non-vanishing vacuum expectation value v labeled by the symbol `vev`,

```
M$vevs = { {phi1[2], vev} }
```

as shown in Sect. 3.3. Then, we choose to implement the mixing of the H^0, R^0 and I^0 fields to the physical h_1, h_2 and h_3 fields in a two-step manner. In a first stage, the gauge eigenstates are split into their scalar and pseudoscalar components,

```
Mix["1p"] == {
  MassBasis  -> { {H0}, {G0} },
  GaugeBasis -> { phi1[2] },
  Value      -> { {{1}}, {{1}} }
}
```

```
Mix["2p"] == {
  MassBasis  -> { {R0}, {I0} },
  GaugeBasis -> { phi2[2] },
  Value      -> { {{1}}, {{1}} }
}
```

following the syntax introduced in Sects. 3.1 and 3.2 and making use of the self-explained symbols `H0` and `R0` (`G0` and `I0`) for representing the (pseudo)scalar degrees of freedom. Similarly, we can employ the mixing infrastructure to map the charged components of ϕ_1 and ϕ_2 to the physical fields G^+ and H^+ by means of a 1×1 identity matrix. Since this procedure is trivial, we omit any further details from the present manuscript and refer to the model implementation [32]. In a second stage, the rotation to the physical fields, represented by the symbols `h1`, `h2` and `h3`, is declared as

```
Mix["1s"] == {
  MassBasis    -> { h1, h2, h3 },
  GaugeBasis   -> { H0, R0, I0 },
  MixingMatrix -> NH,
  BlockName    -> NHMIX
}
```

where we associate the symbol `NH` to the corresponding mixing matrix and assign the Les Houches block (`IM`)`NHMIX` to the numerical value of its elements. The neutral squared mass matrix $\mathcal{M}^2$ can then be derived from the model Lagrangian (represented by the symbol `L2HDM`) by typing, in the MATHEMATICA session,

```
ComputeMassMatrix[L2HDM, Mix->"1s"]
```

As a result, one recovers the well-known expression depending on the most general scalar potential parameters λ_i and μ_i (see Ref. [2] for further information),

$$
\mathcal{M}^2 = \begin{pmatrix} 2\lambda_1 v^2 & \Re[\lambda_6]v^2 & -\Im[\lambda_6]v^2 \\ \Re[\lambda_6]v^2 & m_\pm^2 + [\frac{\lambda_4}{2}+\lambda_5]v^2 & 0 \\ -\Im[\lambda_6]v^2 & 0 & m_\pm^2 + [\frac{\lambda_4}{2}-\lambda_5]v^2 \end{pmatrix},
$$

(8)

after having introduced the squared mass of the charged Higgs boson $m_\pm^2 = 1/2\lambda_3 v^2 + \mu_2$ and removed two of the μ-parameters by means of the potential minimization conditions, $\mu_1 = -\lambda_1 v^2$ and $\mu_3 = -1/2\lambda_6 v^2$.

The numerical value of the unitary matrix U diagonalizing $\mathcal{M}^2$ is obtained by generating and making use of the ASPERGE package, as shown in Sect. 4. We fix, adopting a representative benchmark scenario, the Higgs potential parameters to $\lambda_1 = \lambda_2 = \lambda_3 = 1.0$, $\lambda_4 = 0.5$, $\lambda_5 = 0.4$, $\lambda_6 = 0.3$, $\lambda_7 = 0.2$ and $\mu_2 = 6 \cdot 10^4$ GeV. This leads to the three mass eigenvalues

$$
(m_{h1}, m_{h2}, m_{h3}) = (285, 327, 379)\ \text{GeV},
$$

(9)

whereas the mixing matrix reads

$$
U = \begin{pmatrix} 0 & 0 & -i \\ 0.784 & -0.621 & 0 \\ -0.621 & -0.784 & 0 \end{pmatrix}.
$$

(10)

These results are in good agreement with those obtained by means of the TWOHIGGSCALC calculator [16].

5.2 The minimal left-right symmetric standard model

The LRSM [39–45] is an extension of the Standard Model with an enlarged $SU(3)_c \times SU(2)_L \times SU(2)_R \times U(1)_{B-L}$ gauge symmetry. In this model, the fermionic degrees of freedom of the Standard Model lying in the trivial representation of $SU(2)_L$ are collected into $SU(2)_R$ doublets, as shown in the first part of Table 1 where the model matter field content is presented together with the associated quantum numbers. In addition, the symmetry-breaking mechanism down to electromagnetism is also more involved, relying on an enriched Higgs sector (see the second part of the table).

The LRSM Lagrangian consists of standard kinetic and gauge interaction terms for all fields as well as of the Yukawa interactions

$$
\begin{aligned}
\mathcal{L}_{\text{LR}}^{\text{Y}} = &\ \bar{Q}_L^c \mathbf{y}_{\mathbf{Q}}^{(1)} \hat{\Phi} Q_R + \bar{L}_L^c \mathbf{y}_\ell^{(1)} \hat{\Phi} L_R \\
&+ \bar{Q}_R^c \mathbf{y}_{\mathbf{Q}}^{(2)} \Phi^\dagger Q_L + \bar{L}_R^c \mathbf{y}_\ell^{(2)} \Phi^\dagger L_L \\
&+ \hat{\bar{L}}_L^c \mathbf{y}_\ell^{(3)} \Delta_L L_L + \hat{\bar{L}}_R \mathbf{y}_\ell^{(4)} \Delta_R L_R^c + \text{h.c.}
\end{aligned}
$$

(11)

Table 1 Field content of the LRSM, given together with their representation under the $SU(3)_c \times SU(2)_L \times SU(2)_R \times U(1)_{B-L}$ gauge group. The $SU(2)_L$ ($i, j = 1, 2$) and $SU(2)_R$ ($i', j' = 1, 2$) fundamental index structure is explicitly indicated

Field	Components	Representation
Q_L^i	$\begin{pmatrix} u_L \\ d_L \end{pmatrix}$	$(\mathbf{3}, \mathbf{2}, \mathbf{1}, \frac{1}{3})$
$Q_{Ri'}$	$\begin{pmatrix} u_R^c & d_R^c \end{pmatrix}$	$(\bar{\mathbf{3}}, \mathbf{1}, \mathbf{2}^*, -\frac{1}{3})$
L_L^i	$\begin{pmatrix} \nu_L \\ \ell_L \end{pmatrix}$	$(\mathbf{1}, \mathbf{2}, \mathbf{1}, -1)$
$L_{Ri'}$	$\begin{pmatrix} \nu_R^c & \ell_R^c \end{pmatrix}$	$(\mathbf{1}, \mathbf{1}, \mathbf{2}^*, 1)$
$\Phi^i{}_{i'}$	$\begin{pmatrix} \Phi^0 & \Phi^+ \\ \Phi'^- & \Phi'^0 \end{pmatrix}$	$(\mathbf{1}, \mathbf{2}, \mathbf{2}^*, 0)$
$\Delta_L{}^i{}_j$	$\begin{pmatrix} \frac{1}{\sqrt{2}}\Delta_L^+ & \Delta_L^{++} \\ \Delta_L^0 & -\frac{1}{\sqrt{2}}\Delta_L^+ \end{pmatrix}$	$(\mathbf{1}, \mathbf{3}, \mathbf{1}, 2)$
$\Delta_R{}^{i'}{}_{j'}$	$\begin{pmatrix} \frac{1}{\sqrt{2}}\Delta_R^+ & \Delta_R^{++} \\ \Delta_R^0 & -\frac{1}{\sqrt{2}}\Delta_R^+ \end{pmatrix}$	$(\mathbf{1}, \mathbf{1}, \mathbf{3}, 2)$

In this equation, all indices are understood, the matrices $\mathbf{y}_{\mathbf{Q}}^{(\mathbf{i})}$ and $\mathbf{y}_\ell^{(\mathbf{i})}$ are 3×3 matrices in flavor space and the superscript c indicates charge conjugation.[6] Moreover, gauge invariance is ensured by the introduction of the hatted fields

$$
\hat{\Phi}_i{}^{i'} = \epsilon_{ij}\epsilon^{i'j'}\Phi^j{}_{j'}, \qquad \hat{L}_{Li} = \epsilon_{ij}L_L^j, \qquad \hat{L}_R^{i'} = \epsilon^{i'j'}L_{Rj'},
$$

(12)

where the rank-two antisymmetric tensors with lower and upper indices are defined by $\epsilon_{12} = -\epsilon^{12} = 1$. Introducing Higgs mass parameters μ_i and quartic interaction strengths λ_i, ρ_i and α_i, the scalar potential reads

$$
\begin{aligned}
\mathcal{L}^{\text{H}} = &\ \mu_1^2 \text{Tr}[\Phi^\dagger \Phi] - \lambda_1 (\text{Tr}[\Phi^\dagger \Phi])^2 - \lambda_2 \text{Tr}[\Phi^\dagger \Phi \Phi^\dagger \Phi] \\
&- \frac{1}{2}\lambda_3 (\text{Tr}[\hat{\Phi}\Phi^t] + \text{Tr}[\Phi^\dagger \hat{\Phi}^{\dagger t}])^2 \\
&- \lambda_4 \text{Tr}[\Phi^\dagger \Phi \hat{\Phi}^t \hat{\Phi}^{\dagger t}] \\
&- \frac{1}{2}\lambda_5 (\text{Tr}[\hat{\Phi}\Phi^t] - \text{Tr}[\Phi^\dagger \hat{\Phi}^{\dagger t}])^2 \\
&- \frac{1}{2}\lambda_6 (\text{Tr}[\Phi^\dagger \hat{\Phi}^{\dagger t}\Phi^\dagger \hat{\Phi}^{\dagger t}] + \text{Tr}[\hat{\Phi}^t \Phi \hat{\Phi}^t \Phi]) \\
&+ \mu_2^2 (\text{Tr}[\Delta_L^\dagger \Delta_L] + \text{Tr}[\Delta_R^\dagger \Delta_R]) \\
&- \rho_1 (\text{Tr}[\Delta_L^\dagger \Delta_L]^2 + \text{Tr}[\Delta_R^\dagger \Delta_R]^2)
\end{aligned}
$$

---

[6] We recall that the components of the field Q_R are charge conjugate (see Table 1).

$$- \rho_2 \big(\mathrm{Tr} \big[\Delta_L^\dagger \Delta_L \Delta_L^\dagger \Delta_L \big] + \mathrm{Tr} \big[\Delta_R^\dagger \Delta_R \Delta_R^\dagger \Delta_R \big] \big)$$
$$- \rho_3 \mathrm{Tr} \big[\Delta_L^\dagger \Delta_L \big] \mathrm{Tr} \big[\Delta_R^\dagger \Delta_R \big]$$
$$- \alpha_1 \mathrm{Tr} \big[\Phi^\dagger \Phi \big] \big(\mathrm{Tr} \big[\Delta_L^\dagger \Delta_L \big] + \mathrm{Tr} \big[\Delta_R^\dagger \Delta_R \big] \big)$$
$$- \alpha_2 \big(\mathrm{Tr} \big[\Delta_R^\dagger \Phi^\dagger \Phi \Delta_R \big] + \mathrm{Tr} \big[\Delta_L^\dagger \Phi \Phi^\dagger \Delta_L \big] \big)$$
$$- \alpha_3 \big(\mathrm{Tr} \big[\Delta_R^\dagger \hat{\Phi}^t \hat{\Phi}^{\dagger t} \Delta_R \big] + \mathrm{Tr} \big[\Delta_L^\dagger \hat{\Phi}^{\dagger t} \hat{\Phi}^t \Delta_L \big] \big). \quad (13)$$

Since the corresponding FEYNRULES model description is standard, we refer to the FEYNRULES manual [1] and leave all implementation details out of this work.

In the LRSM, the symmetry-breaking mechanism is performed in two steps. At high energy, the $SU(2)_L \times SU(2)_R \times U(1)_{B-L}$ gauge symmetry is spontaneously broken to the electroweak symmetry, the latter being subsequently broken to electromagnetism at a lower scale. Consequently, the neutral components of the scalar fields get vacuum expectation values at the minimum of the potential, $\langle \Phi^0 \rangle = v/\sqrt{2}$, $\langle \Phi'^0 \rangle = v'/\sqrt{2}$ and $\langle \Delta_{L,R}^0 \rangle = v_{L,R}/\sqrt{2}$, by which they are shifted. In the rest of this section, we focus on the mixing of the neutral Higgs fields and illustrate the way to implement a two-stage field rotation. For all the other mixing relations of the LRSM, we refer to the implementation [32].

We first assume, motivated by neutrino mass and kaon system data [45, 46], that $v_L = v' \approx 0$. Next, we implement the rotation associated with the diagonalization of the third generator of $SU(2)$ in the adjoint representation as

```
Mix["2a"] == {
  MassBasis  -> { DLpp, DL0 },
  GaugeBasis -> { DL[1], DL[2] },
  Value      -> { {1/Sqrt[2], -I/Sqrt[2]},
                  {1/Sqrt[2],  I/Sqrt[2]}}
}
```

depicting the example of the $SU(2)_L$ Higgs triplet. These replacement rules translate the rotation of the Δ_L^1 and Δ_L^2 fields, represented by the DL[1] and DL[2] symbols, to the Δ^0 and Δ^{++} states labeled by DL0 and DLpp. Then, the neutral fields Δ_L^0, Δ_R^0, Φ^0 and Φ'^0, represented by the symbols DL0, DR0, phi[1,1] and phi[2,2], mix to four scalar degrees of freedom h_1^0, h_2^0, h_3^0 and h_4^0, two physical pseudoscalar Higgs bosons a_1^0 and a_2^0 and two Goldstone bosons G_1^0 and G_2^0 to be eaten by the Z and Z' vector fields when getting massive. Introducing the corresponding symbols h01, h02, h03, h04, a01, a02, G01 and G02, these rotations are implemented as

```
Mix["2e"] == {
  MassBasis    -> { {h01,h02,h03,h04},
                    {G01,G02,a01,a02} },
  GaugeBasis   ->
    { DL0,DR0,phi[1,1],phi[2,2] },
  MixingMatrix -> { UHN,UAN },
  BlockName    -> { HMIX,AMIX}
}
```

The two symbols UHN and UAN respectively denote the scalar and pseudoscalar mixing matrices, the numerical value of their elements being included in the two Les Houches blocks (IM)HMIX and (IM)AMIX.

Typing, in MATHEMATICA, the commands

```
ComputeMassMatrix[Lag, Mix->"2e"]
MassMatrix["2e", "S"]
```

allows to calculate both the scalar and pseudoscalar mass matrices and display the scalar squared mass matrix $\tilde{\mathcal{M}}^2$ to the screen. It reads,

$$\tilde{\mathcal{M}}^2 = \begin{pmatrix} A & 0 & 0 & 0 \\ 0 & B & (\alpha_1 + \alpha_3)vv_R & 0 \\ 0 & (\alpha_1 + \alpha_3)vv_R & C & 0 \\ 0 & 0 & 0 & D \end{pmatrix}, \quad (14)$$

where we have introduced the quantities

$$\begin{aligned} A &= \frac{1}{2}(\alpha_1 + \alpha_3)v^2 - \mu_2^2 + \frac{1}{2}\rho_3 v_R^2, \\ B &= \frac{1}{2}(\alpha_1 + \alpha_3)v^2 - \mu_2^2 + 3(\rho_1 - \rho_2)v_R^2, \\ C &= 3(\lambda_1 + \lambda_2)v^2 + \frac{1}{2}(\alpha_1 + \alpha_3)v_R^2 - \mu_1^2, \\ D &= (\lambda_1 + 4\lambda_3 + \lambda_4 + \lambda_6)v^2 + \frac{1}{2}(\alpha_1 + \alpha_2)v_R^2 - \mu_1^2. \end{aligned} \quad (15)$$

In order to numerically compute the unitary matrix U diagonalizing $\tilde{\mathcal{M}}^2$, we use the ASPERGE program, generated from FEYNRULES by issuing

```
WriteAsperge[Lag]
```

after fixing the external Lagrangian parameters of the model to $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_6 = 0.1$, $\alpha_1 = 0.1$, $\alpha_2 = 0.3$, $\alpha_3 = 0.1$, $\rho_1 = 0.1$, $\rho_2 = 0$ and $\rho_3 = 0.5$. In addition, the μ-terms are deduced from the minimization conditions of the scalar potential and the vacuum expectation values are taken as $v = 248$ GeV and $v_R = 6000$ GeV. It should be noted that the relevance of such numerical values is going beyond the scope of this paper, and could be addressed by means of external packages such as the one presented in Ref. [47]. Once ASPERGE is executed, one obtains

$$U = \begin{pmatrix} 0 & -0.041 & 0.99 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0.99 & 0.041 & 0 \end{pmatrix}, \quad (16)$$

the mass eigenvalues being

$$(m_1, m_2, m_3, m_4) = (111, 1905, 2324, 2686) \text{ GeV}. \quad (17)$$

5.3 The minimal supersymmetric standard model

In supersymmetric extensions of the Standard Model, each of the model's degrees of freedom comes accompanied by a superpartner with opposite statistics. The minimal version of such theories, the so-called MSSM [48, 49], has been originally implemented in FEYNRULES by making use of its superspace module [4]. We hence refer, on the one hand, to Ref. [4] for notations and conventions and, on the other hand, to the new model implementation [32] for more information on the way in which particle mixings have been implemented. In the rest of this subsection, we employ the MSSM implementation to illustrate fermion mixing declaration.

After electroweak symmetry breaking, the left-handed, two-component Weyl fermionic, gaugino and higgsino fields $\tilde{W}^{\pm}$ and $\tilde{H}_d^-/\tilde{H}_u^+$ mix to the chargino eigenstates $\chi^{\pm}$. Introducing the two rotation matrices U and V (labeled by the symbols UU and VV), this mixing is declared through an instance of the mixing class,

```
Mix["3d"]=={
  MassBasis   -> { {chmw[1],chmw[2]},
                   {chpw[1],chpw[2]} },
  GaugeBasis  -> { {wowm,hdw[2]},
                   {wowp,huw[1]} },
  BlockName   -> {UMIX,VMIX},
  MixingMatrix -> {UU,VV}
}
```

In this list of replacement rules, wowm (hdw) and wowp (huw) are the labels of the negatively and positively charged wino (higgsino) states, the related mass eigenstates being represented by the symbols chmw and chpw. In addition, we link the mixing matrices U and V to the Les Houches blocks UMIX and VMIX.

Like in the previous subsections, we employ FEYN-RULES to extract the corresponding tree-level mass matrix $\mathcal{M}'$ from the Lagrangian,

$$\mathcal{M}' = \begin{pmatrix} M_2 & \sqrt{2}m_W \sin\beta \\ \sqrt{2}m_W \cos\beta & \mu \end{pmatrix}, \tag{18}$$

where m_W denotes the W-boson mass, μ the superpotential Higgs mixing parameter, M_2 the supersymmetry-breaking wino mass and $\tan\beta$ is defined as the ratio of the two neutral Higgs field vacuum expectation values $\tan\beta = \langle H_u^0/H_d^0 \rangle$.

As in the original implementation, we choose the typical minimal supergravity point SPS 1a [50] as a benchmark scenario. Then, we generate the MSSM ASPERGE program and use it to compute numerically the mixing matrices U and V,

$$U = \begin{pmatrix} 0.918 & -0.397 \\ -0.397 & -0.918 \end{pmatrix}, \qquad V = \begin{pmatrix} 0.974 & -0.226 \\ -0.226 & -0.974 \end{pmatrix}, \tag{19}$$

as well as the corresponding mass eigenvalues

$$(m_{\chi_1^{\pm}}, m_{\chi_2^{\pm}}) = (176, 382) \text{ GeV}. \tag{20}$$

Those values are in good agreement with those returned by commonly used MSSM spectrum generators.

6 Summary

In this paper, we have presented an extension of the FEYN-RULES package dedicated to the automated generation of the particle mass spectrum and mixing structure associated to any Lagrangian-based quantum field theory. The new module is based on the introduction of a new structure for particle mixing declaration allowing, on the one hand, for the analytical computation of all the model mass matrices, and, on the other hand, for the generation of a C++ program dubbed ASPERGE, yielding the numerical evaluation of the associated rotation matrices. We illustrate the strength of this new module in the context of the Two-Higgs-Doublet Model, the Minimal Left-Right Symmetric Standard Model and the Minimal Supersymmetric Standard Model.

References

1. N.D. Christensen, C. Duhr, Comput. Phys. Commun. **180**, 1614 (2009)
2. 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)
3. N.D. Christensen, C. Duhr, B. Fuks, J. Reuter, C. Speckner, Eur. Phys. J. C **72**, 1990 (2012)
4. C. Duhr, B. Fuks, Comput. Phys. Commun. **182**, 2404 (2011)
5. B. Fuks, Int. J. Mod. Phys. A **27**, 1230007 (2012)

6. A. Pukhov, E. Boos, M. Dubinin, V. Edneral, V. Ilyin, D. Kovalenko, A. Kryukov, V. Savrin et al., hep-ph/9908288
7. E. Boos et al., Nucl. Instrum. Methods A **534**, 250 (2004)
8. A. Pukhov, hep-ph/0412191
9. A. Belyaev, N.D. Christensen, A. Pukhov, arXiv:1207.6082 [hep-ph]
10. T. Hahn, M. Perez-Victoria, Comput. Phys. Commun. **118**, 153 (1999)
11. T. Hahn, Comput. Phys. Commun. **140**, 418 (2001)
12. T. Hahn, PoS ACAT **08**, 121 (2008)
13. S. Agrawal, T. Hahn, E. Mirabella, J. Phys. Conf. Ser. **368**, 012054 (2012)
14. T. Stelzer, W.F. Long, Comput. Phys. Commun. **81**, 357 (1994)
15. F. Maltoni, T. Stelzer, J. High Energy Phys. **0302**, 027 (2003)
16. J. Alwall, P. Demin, S. de Visscher, R. Frederix, M. Herquet, F. Maltoni, T. Plehn, D.L. Rainwater et al., J. High Energy Phys. **0709**, 028 (2007)
17. J. Alwall, P. Artoisenet, S. de Visscher, C. Duhr, R. Frederix, M. Herquet, O. Mattelaer, AIP Conf. Proc. **1078**, 84 (2009)
18. J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, T. Stelzer, J. High Energy Phys. **1106**, 128 (2011)
19. T. Gleisberg, S. Hoeche, F. Krauss, A. Schalicke, S. Schumann, J.-C. Winter, J. High Energy Phys. **0402**, 056 (2004)
20. T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, J. Winter, J. High Energy Phys. **0902**, 007 (2009)
21. M. Moretti, T. Ohl, J. Reuter, hep-ph/0102195
22. W. Kilian, T. Ohl, J. Reuter, Eur. Phys. J. C **71**, 1742 (2011)
23. C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer, T. Reiter, Comput. Phys. Commun. **183**, 1201 (2012)
24. P. de Aquino, W. Link, F. Maltoni, O. Mattelaer, T. Stelzer, Comput. Phys. Commun. **183**, 2254 (2012)
25. E. Conte, B. Fuks, G. Serret, Comput. Phys. Commun. **184**, 222 (2013)
26. G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, Eur. Phys. J. C **72**, 1889 (2012)
27. G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, J. Phys. Conf. Ser. **368**, 012056 (2012)
28. M. Bahr, S. Gieseke, M.A. Gigg, D. Grellscheid, K. Hamilton, O. Latunde-Dada, S. Platzer, P. Richardson et al., Eur. Phys. J. C **58**, 639 (2008)
29. J. Beringer et al. (Particle Data Group Collaboration), Phys. Rev. D **86**, 010001 (2012)
30. P.Z. Skands, B.C. Allanach, H. Baer, C. Balazs, G. Belanger, F. Boudjema, A. Djouadi, R. Godbole et al., J. High Energy Phys. **0407**, 036 (2004)
31. B.C. Allanach, C. Balazs, G. Belanger, M. Bernhardt, F. Boudjema, D. Choudhury, K. Desch, U. Ellwanger et al., Comput. Phys. Commun. **180**, 8 (2009)
32. http://feynrules.irmp.ucl.ac.be/wiki/ASperGe
33. T. Hahn, physics/0607103
34. G. Belanger, N.D. Christensen, A. Pukhov, A. Semenov, Comput. Phys. Commun. **182**, 763 (2011)
35. G.C. Branco, L. Lavoura, J.P. Silva, Int. Ser. Monogr. Phys. **103**, 1 (1999)
36. I.F. Ginzburg, M. Krawczyk, Phys. Rev. D **72**, 115013 (2005)
37. S. Davidson, H.E. Haber, Phys. Rev. D **72**, 035004 (2005). Erratum-ibid. D **72**, 099902 (2005)
38. H.E. Haber, D. O'Neil, Phys. Rev. D **74**, 015018 (2006)
39. J.C. Pati, A. Salam, Phys. Rev. D **10**, 275 (1974). Erratum-ibid. D **11**, 703 (1975)
40. R.N. Mohapatra, J.C. Pati, Phys. Rev. D **11**, 566 (1975)
41. R.N. Mohapatra, J.C. Pati, Phys. Rev. D **11**, 2558 (1975)
42. G. Senjanovic, R.N. Mohapatra, Phys. Rev. D **12**, 1502 (1975)
43. R.N. Mohapatra, F.E. Paige, D.P. Sidhu, Phys. Rev. D **17**, 2462 (1978)
44. G. Senjanovic, Nucl. Phys. B **153**, 334 (1979)
45. C.S. Lim, T. Inami, Prog. Theor. Phys. **67**, 1569 (1982)
46. R.N. Mohapatra, G. Senjanovic, Phys. Rev. D **23**, 165 (1981)
47. R. Coimbra, M.O.P. Sampaio, R. Santos, arXiv:1301.2599 [hep-ph]
48. H.P. Nilles, Phys. Rep. **110**, 1 (1984)
49. H.E. Haber, G.L. Kane, Phys. Rep. **117**, 75 (1985)
50. B.C. Allanach, M. Battaglia, G.A. Blair, M.S. Carena, A. De Roeck, A. Dedes, A. Djouadi, D. Gerdes et al., Eur. Phys. J. C **25**, 113 (2002)