SARAH

A Model Builder's Tool

Version 3.0 beta

Florian Staub
Institut für Theoretische Physik und Astrophysik, Würzburg
florian.staub@physik.uni-wuerzburg.de

March 4, 2011

Disclaimer
SARAH is published under the GNU library public license [1], this means that you can use it for free. We have tested this software and its result, but we can't guarantee that this software works correctly or that the physical results derived using this software are correct.

We have the following requests:

- If you find any bug, please inform us by eMail: florian.staub@physik.uni-wuerzburg.de
- If you have any suggestions, what is missing or can be improved, please let us know.
- If you use this software, please cite arXiv:0806.0538

[1] http://www.fsf.org/copyleft/lgpl.html

Contents

Chapter 1

Introduction

Supersymmetry (SUSY) is one of the most popular extensions of the standard model (SM) of particle physics [1, 2, 3, 4, 5, 6]: it solves the hierarchy problem [7, 8], leads to unification of the three gauge couplings [9, 10, 11, 12] and offers often a candidate for dark matter [13].
The minimal supersymmetric standard model (MSSM) is nowadays well studied. Every event generator or diagram calculator can handle the MSSM out of the box. Unfortunately, there remains a lot of work if somebody wants to change the supersymmetric model, e.g. extend the particle content, add new gauge groups or add new interactions to the superpotential. First, it must be checked that the new model is free from gauge anomalies. As second step, the full Lagrangian must be derived and all interactions have to be extracted. This is complicated by the fact that the fields in gauge eigenstates have to be rotated to new mass eigenstates: the rotations must be incorporated, mass matrices have to be calculated and diagonalized. Finally, the tadpole equations are needed to find the minimum of the potential. All these steps are needed just to get a rough impression of the new model. If also phenomenological studies should be made by using one of the existing programs, model files must be created. Moreover, for the embedding of the model in a GUT theory, the Renormalization Group Equations (RGEs) are needed. Furthermore, often loop corrections to the masses are demanded.
This is exactly that kind of work `SARAH` was written for. `SARAH` just needs the gauge structure, particle content and superpotential to produce all information about the gauge eigenstates. As gauge groups, all $SU(N)$ groups can be handled and the superfields can transform as any arbitrary, irreducible representation of these groups. Breaking of gauge symmetries and mixings of particles can easily be added. Also the gauge fixing terms can be specified, and the corresponding ghost interactions are calculated automatically. The two-Loop RGEs for the superpotential parameters, the gauge couplings and the soft-breaking parameters are derived. In addition, the self energies are calculated at one-loop level. `SARAH` can write all information about the model to LaTeXfiles, or create model files for `FeynArts` [14] and `CalcHep`/`CompHep` [15, 16], which can also be used for dark matter studies using MicrOmegas [17].
Starting with the third version, `SARAH` is also supposed to be the first 'spectrum-generator-generator': it uses all analytical expressions to generate source code for `SPheno` [18]. The source code can be used to calculate the mass spectrum with `SPheno` for a new model using 2-loop RGEs and 1-loop corrections to the masses. In addition, the necessary routines for two- and three-body decays are written.
The intention by the development of `SARAH` was to make it very flexible: there is a big freedom for the matter and gauge sector which can be handled. The work with `SARAH` should be easy: every information `SARAH` needs are specified in an easy to modify model file. Nevertheless, `SARAH` is also fast: a existing model can be changed within minutes, and the needed time for doing all necessary calculations and writing a model file is normally less than 10 minutes.
This manual is structured as follows: in the next chapter we explain the installation and the general setup

of SARAH. In the third chapter, the commands of SARAH are explained before in two following chapters all possibilities of defining models and parameters are shown in detail. The last two chapters discuss the possible output and the usage of numerical values.

Chapter 2

Quick start

2.1 Download and installation

SARAH is a package for Mathematica [1] and was tested with versions 5.2 and 7.0.
SARAH can be downloaded from

http://theorie.physik.uni-wuerzburg.de/~fnstaub/sarah.html

The package archive contains the following directories:

a) Models: Definition of the different models

b) Package: All package files.

c) LaTeX-Packages: LaTeXpackages, which might be needed for the output of SARAH

During the work, also the directory

Output

is created. It will contain all files written by SARAH.
In addition, the root directory of SARAH contains this manual (sarah.pdf), an overview of all models inlcuded in the package (models.pdf), a short introduction (Readme.txt) as well as an example for the evaluation of the MSSM (Example.nb).
The package should be extracted to the application directory of Mathematica. This directory is Linux

home/user/.Mathematica/Applications/

and

Mathematica-Directory\AddOns\Applications\

in Windows.

[1] Mathematica is a protected product by Wolfram Research

2.2 Run SARAH

After the installation, the package is loaded in Mathematica via

```
<<"sarah-2.0/SARAH.m"
```

and a supersymmetric model is initialized by

```
Start["Modelname"];
```

Here, `Modelname` is the name of the corresponding model file, e.g. for the minimal supersymmetric standard model the command would read

```
Start["MSSM"];
```

or for the next-to-minimal supersymmetric model in CKM basis

```
Start["NMSSM","CKM"];
```

is used. In the following, we refer for all given examples the model file of the MSSM. Our conventions concerning the fields definitions and rotations in the MSSM are given in app. F.

2.3 What happens automatically

When a model is initialized using the `Start` command, this model is first checked for gauge anomalies and charge conservation. If not all checks are fulfilled, a warning is printed. More information about the different checks is given in app. A. Afterwards, the calculation of the complete Lagrangian at tree level starts. The performed steps are presented in app. B.
The next steps are to accomplish all necessary rotations and redefinition of fields: if a gauge symmetry is broken, the fields responsible for the symmetry breaking are getting a vacuum expectation value (VEV) and the gauge fields are rotated. Afterwards, the matter particles are rotated to the new mass eigenbasis and the tadpole equations are derived. These steps can be repeated if more rotations or symmetry breakings are necessary.
During this evaluation some more things might done automatically: particles are integrated out to get an effective theory, the ghost interactions are derived from the gauge fixing part of the Lagrangian or the parametrization of defined interactions is changed. At the end, `SARAH` calculates for all appearing particles in the model the masses at tree level and splits the Lagrangian in different pieces to increase the speed of following calculations. Additionally, if numerical values for all parameters are provided, e.g. in the input files or in a LesHouches spectrum file [19], `SARAH` calculates the eigenvalues of the mass matrices and the rotation matrices.

2.4 Commands

The most important commands to work with `SARAH` are:

a) `Vertex[Fields, Options]`: Calculates a vertex for given fields

b) `MassMatrices[States]`: Shows all mass matrices for given eigenstates `States`

c) `TadpoleEquations[States]`: Shows all tadpole equations for given eigenstates `States`

d) `MassMatrix[Field]`: Shows the mass matrix of the field `field`

e) `TadpoleEquation[Field]`: Shows the tadpole equation corresponding to the VEV `VEV`

f) `CalcRGEs[Options]`: Calculates the RGEs

g) `CalcLoopCorrections[Options]`: Calculates one-loop corrections for given eigenstates `States`

h) `ModelOutput[States,Options]`: Create output defined by options for given eigenstates `States`.

i) `MakeVertexList[States,Options]`: Calculates all vertices for given eigenstates `States`

j) `MakeSPheno[Options]`: Writes source code for `SPheno`

k) `MakeTeX[Options]`: Writes LaTeXfiles

l) `MakeCHep[Options]`: Writes `CalcHep`/`CompHep` model files

m) `MakeFeynArts`: Writes `FeynArts` model file

n) `MakeWHIZARD[Options]`: Writes model files for `WHIZARD` and `Omega`

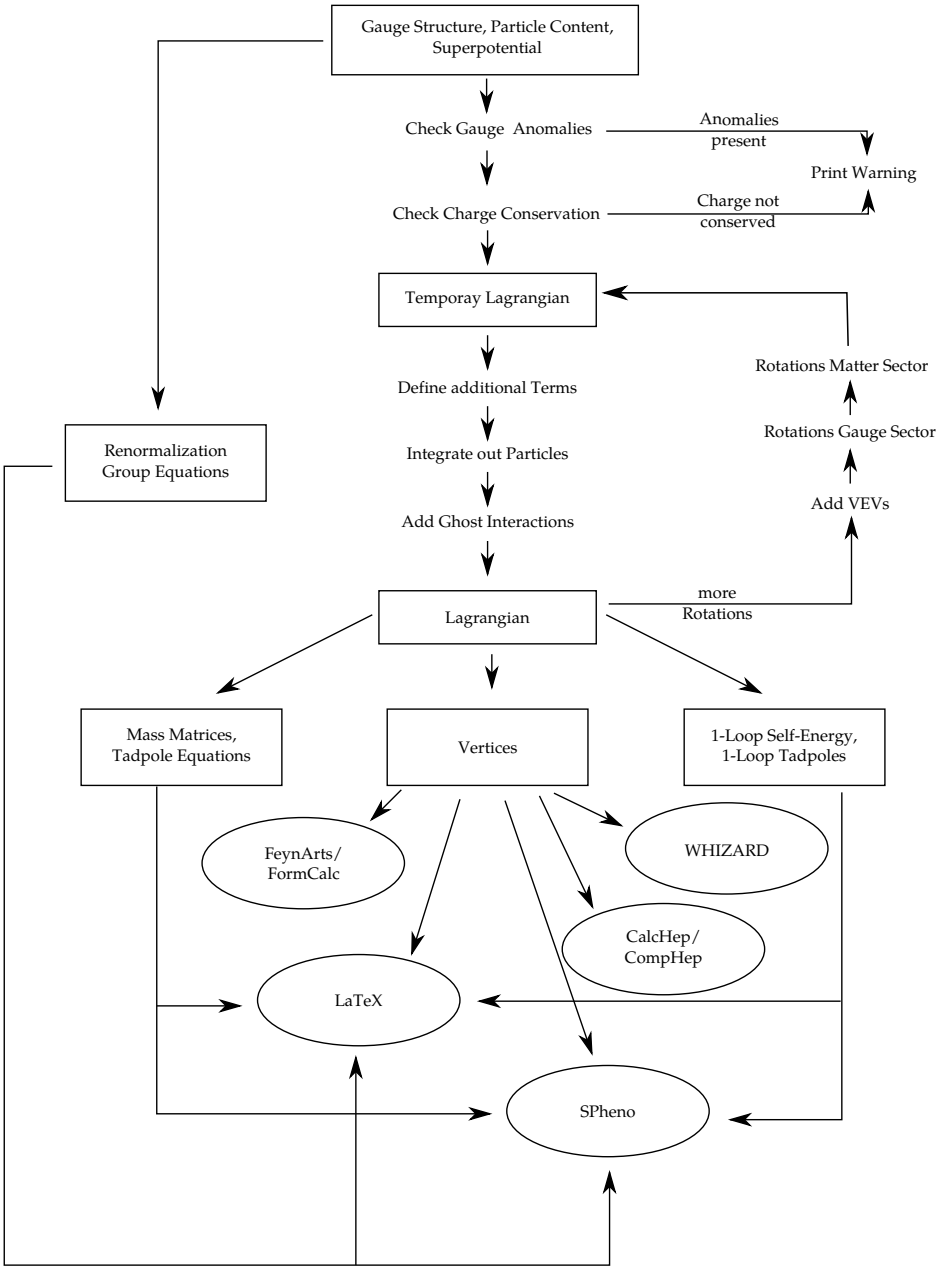

Figure 2.1: Setup of SARAH

Chapter 3

Working with SARAH

3.1 Definition of particles

Before we explain the different functions of SARAH, we have to clarify the nomenclature. All gauge eigenstates are named as follows:

```
ParticleType <> Basis <> [Indices]
```

Type Here, ParticleType is just one letter and indicates the type of a field. The convention is as follows:

a) F for fermionic component of chiral superfield

b) S for scalar component of chiral superfield

c) f for fermionic component of vector superfield

d) V for bosonic component of vector superfield

e) g for ghost field

In addition, there are two types of auxiliary fields. This auxiliary field are not related in any way to the auxiliary components of the superfields in SUSY theories, but they are needed only for writing a CalcHep/CompHep model file (see sec. 6.3.3). The type indicating letters are:

a) A for an auxiliary scalar

b) a for an auxiliary vector boson

> After rotating gauge fields to new mass eigenstates, there are no longer constraints concerning the names of fermions and scalars. However, vector bosons must still begin with V because the corresponding ghosts are automatically added!

Basis The Basis of a particle is the name of the underlying superfield. In the MSSM for example, this can be dL for the superfield of the left down quarks and squarks or G for the vector superfield transforming under the strong interaction.

Indices `Indices` is the list of the indices which the particle carries. There are three different kinds of indices:

a) `generation`: For all particles which appear in more than one generation

b) `lorentz:` For all particles carrying a Lorentz index

c) `charge:` For all particles charged under a non-abelian gauge group if this indices are not implicit

Examples To clarify the above definitions, here some examples:

a) `VB[{lorentz}]`: B-Boson with one Lorentz index

b) `fB`: Bino

c) `SHd0`: Neutral down Higgs

d) `FHd0`: Neutral down Higgsino

e) `VG[{generation,Lorentz}]`: Gluon with one generation and one Lorentz index

f) `fG[{generation}]`: Gluino with one generation index

g) `gG[{generation}]`: Gluon ghost with one generation index

h) `SdL[{generation,color}]`: Left handed d-squark with one generation and one color index

i) `FdL[{generation,color}]`: Left handed d-quark with one generation and one color index

j) `hh[{generation}]`: Neutral, CP-even Higgs (light and heavy Higgs) after EWSB

A comprehensive overview about all models defined in
SARAH is given in the file `Models.pdf`. Furthermore, the MSSM is discussed in app. F.

3.1.1 Antiparticles

There are two function to assign antiparticles: depending on the type of the particle `conj` or `bar` are used.

a) Scalar, vector boson and Weyl spinor: `conj`, e.g. `conj[SdR]` or `conj[VWm]`.

b) Dirac fermion and ghost: `bar`, e.g `bar[Fd]` or `bar[gG]`.

`SARAH` checks if a particle is real or complex or a Majorana fermion and simplifies the expressions by using

```
conj[RP] := RP
bar[MF] := MF
```

for a real particle `RP` or real parameter, and for a Majorana fermion `MF`. The name of all Majorana fermions of the current model is saved in the list `MajoranaPart`, and all real parameters and particles are listed in `realVar`.
Some words about the necessity of `conj`: The function `conj` is very similar to the existing function `Conjugate` of Mathematica. But at least with Mathematica 5.2 there are some problems concerning this function: it is not possible to calculate the derivative with respect to a complex conjugate variable, and `Conjugate` is a numerical function. Hence, it is in some cases too slow in handling big analytical expressions. This was improved in Mathematica 7.0 but we want to stay compatible also with version 5.2.

3.2 Different eigenstates of one model

While calculating the Lagrangian of a model, SARAH saves the information of the different eigenstates of the model. The name of the eigenstates can be defined by the user in the model file, see (5.3). For the model file included in the official package, the eigenstates are

a) GaugeES: gauge eigenstates without any rotation

b) EWSB: eigenstates after electroweak symmetry breaking

c) SCKM: eigenstates in Super-CKM basis

d) TEMP: auxiliary eigenstates with no physical meaning

3.3 Model information

There is a lot of information automatically calculated and saved by SARAH. In this section is shown, how this information can be accessed and used.

3.3.1 Particle content

To get an overview of all particles of the different eigenstates, use

```
Particles[Eigenstates]
```

e.g. `Particles[GaugeES]` or `Particles[EWSB]` for the gauge eigenstates or the eigenstates after electroweak symmetry breaking (EWSB), respectively. The output is a list with the following information about each particle:

a) Name of the particle

b) Type of the particle (`F` for fermion, `S` for scalar, `G` for ghosts, `A` for auxiliary field)

c) Number of first generation (can be different from 1 in effective theories)

d) Number of last generation

e) Indices of the particle

Example For instance, the entry for the gauge eigenstates of the left-down quark reads

```
{FdL, 1, 3, F, {generation, color}}
```

3.3.2 Masses of particles

SARAH automatically calculates the tree level masses of all particles. This information is saved in

```
Masses["Name of Eigenstates"]
```

It returns are a list with replacements rules. First, a statement `Mass[]` with the name of the particle is given followed by the value. There are three possibilities for the value

a) Expression: if a expression is used for a mass, SARAH has calculated the tree level mass depending on other parameters of the model.

b) `MassGiven`: this means that a numerical value for the mass was given in the particle file (see 4.3).

c) `MassRead`: this means that the value of the mass was read in from a LesHouches input file.

More information of defining masses is given in section 4.3.

Example The mass of the Z-Boson (`VZ`) after EWSB is saved in `Masses[EWSB]`. The corresponding entry is

```
Mass[VZ] -> ((vd^2 + vu^2)*(g2*Cos[ThetaW] + g1*Sin[ThetaW])^2/4
```

3.3.3 Mass Matrices

`SARAH` calculates automatically the mass matrices before rotating the fields to the new eigenstates and saves the information in arrays. The basis of the rotations can be seen by using

```
MixBasis["Name of Eigenstates"]
```

The matrices itself are saved in two arrays:

```
MassMatrices["Name of Eigenstates"]
```

and

```
MassMatricesFull["Name of Eigenstates"]
```

The difference between this two arrays is that in the first one, the different generations are written as indices, while in the second on the generation indices are explicitly inserted. This means, in the first case the basis for the mass matrix in the down squark sector is just

```
(SdL[{gn,cn}],SdL[{gm,cm}])
```

while in the second case the basis vector is

```
(SdL[{1,cn1}],SdL[{2,cn2}],SdL[{3,cn3}],SdL[{1,cm1}],SdL[{2,cm2}],SdL[{3,cm2}])
```

It is also possible to use the command

```
MassMatrix[Field]
```

to obtain the mass matrix for a specific particle.

Example Let's have a look at the down-squark sector after EWSB.

```
MixBasis[EWSB][[1]]
```

returns the basis of the mass matrix:

```
{{SdL, SdR}, {conj[SdL], conj[SdR]}}
```

The (1,1) element of the mass matrix is saved in

```
MassMatricesFull[EWSB][[1,1,1]]
```

and looks like

```
-(g1^2*vd^2)/24 - (g2^2*vd^2)/8 + (g1^2*vu^2)/24 +
   (g2^2*vu^2)/8 + mq2[1, 1] + (vd^2*Yd[1, 1]^2)/2
```

The same result is obtained by

```
MassMatrix[Sd][[1,1]]
```

3.3.4 Tadpole Equations

Gauge symmetries are broken, if particles receive a VEVs. These VEVs v_i should minimize the potential and therefore fulfill the equations

$$\frac{\partial V}{\partial v} = 0 \tag{3.1}$$

V is the scalar potential of the model. Eq. (3.1) are the so called tadpole equations which are cubic equations in the VEVs. SARAH saves the left hand side of (3.1) in arrays named

```
TadpoleEquations["Name of Eigenstates"]
```

The order of the tadpole equations in this array corresponds to the order of the definition of VEVs in the model file, see (5.3.1.1).
There is also the shorter command

```
TadpoleEquation[VEV]
```

to obtain the tadpole equations corresponding to a specific VEV.

Example The tadpole equation for v_d after EWSB is saved in

```
TadpoleEquations[EWSB][[1]]
```

and reads

```
mHd2*vd + (g1^2*vd^3)/8 + (g2^2*vd^3)/8 - (g1^2*vd*vu^2)/8 -
  (g2^2*vd*vu^2)/8 + vd*\[Mu]^2 - vu*B[\[Mu]]
```

The same result can be obtained by

```
TadpoleEquation[vd]
```

3.3.5 Parts of the Lagrangian

The most important, but also most complex piece of a model, is the full Lagrangian. SARAH calculates the full Lagrangian from the superpotential and the gauge sector by using the method explained in appendix B. The final results, the Lagrangians for the different eigenstates are saved as

```
Lagrangian["Name of Eigenstates"]
```

For a realistic SUSY model, the Lagrangian is generally very lengthy. Therefore, SARAH splits it in different parts in order to speed up some calculations. This splitting might be also helpful for analyzing the structure of interactions involving different kinds of fields. An overview of all names for the different parts of the Lagrangian is given in C.
Also the results of the different steps during then calculation of the Lagrangian in gauge eigenstates (e.g. F-Terms, D-Terms, kinetic parts) are saved. Thus, it is possible to have a detailed look at specific parts of the Lagrangian. The names of the parts are also given in appendix B.

3.3.6 Writing all information about particles and parameters in an external file

All information about particles and parameters of the considered model can by written in two files `ParticleInfo.m` and `ParameterInfo`, which are saved in

```
../\SARAH/Output/"Model Name"/
```

by using

```
ExportModelInformation;
```

The file `ParticleInfo.m` contains the following information for all eigenstates of the model

a) R-Parity

b) PDG

c) LaTeXname

d) Output name

e) `FeynArts` number

f) Type

g) Self-conjugated or not

h) Number of generations

i) Indices

while `ParameterInfo.m` contains the following information about all parameters

a) Dependence on other parameters

b) Real or complex

c) Numerical value

d) Position in LesHouches input

e) LaTeX name

3.4 Calculating Vertices

One of the main functions of SARAH is to calculate the vertices for a model. In contrast to the most other calculations, vertices normally are not calculated automatically: it can last several minutes to calculate all vertices of a model sometimes these calculations are not necessary. Of course, it is also possible to tell SARAH that all vertices should automatically calculated, see section 6.5. In this section, we want to focus on calculating vertices 'by hand'.
Calculating vertices is done with

```
Vertex[ParticleList,Options]
```

`ParticleList` is a list containing the involved fields. This list can consist of up to 6 particles if an effective theory is analyzed (see 5.4).
The following `Options` are supported by the `Vertex` command:

a) `Eigenstates`, Value: `Name of Eigenstates`, Default: Last entry in `NameOfState`
Fixes the considered eigenstates

b) `UseDependences`: Value `True` or `False`, Default: `False`
Optional relations between the parameters (see section 4.2) will be used, if `UseDependences` is set to `True`.

The output of `Vertex` is an array:

```
{{ParticleList},{{Coefficient 1, Lorentz 1},{Coefficient 2, Lorentz 2},...}
```

First, the list of the involved particles is given and the indices are inserted. The second part consists of the value of the vertex and can be also a list, if different Lorentz structures are possible. In the part independent of Lorentz indices can appear

a) `Delta[a,b]`: Kronecker delta $\delta_{\alpha\beta}$

b) `ThetaStep[i,j]`: Step function Θ_{ij}

c) `Lam[t,a,b]`: Gell-Mann matrix $\lambda^t_{\alpha\beta}$

d) `LambdaProd[x,y][a,b]`: Matrix product of two Gell-Mann matrices $\left(\lambda^x\lambda^y\right)_{\alpha\beta}$

e) `Sig[t,a,b]`: Pauli matrix $\sigma^t_{\alpha\beta}$

f) `SigmaProd[x,y][a,b]`: Matrix product of two Pauli matrices $\left(\sigma^x\sigma^y\right)_{\alpha\beta}$

g) `fSU3[i,j,k]`: Structure constants of $SU(3)$: f^{ijk}

h) `fSU2[i,j,k]`: Structure constants of $SU(2)$: ϵ^{ijk}

i) `FST[SU[N]][i,j,k]`: Structure constants of $SU(N)$: f_N^{ijk}

j) Couplings, e.g. `g1`, `g2`, `g3`, `Ye[a,b]`, `Yd[a,b]`, `Yu[a,b]`, ...

k) Mixing matrices, e.g. `ZD[a,b]`, ...

The part transforming under the Lorentz group can consist of

a) `gamma[lor]`: Gamma matrix γ_μ

b) `g[lor1,lor2]`: Metric tensor $g_{\mu\nu}$

c) `Mom[particle,lor]`: Momentum p^μ_P of particle P

d) `PL`, `PR`: Polarization operators $P_L = \frac{1-\gamma_5}{2}$, $P_R = \frac{1+\gamma_5}{2}$

e) `1`: If the vertex is a Lorentz scalar.

f) `LorentzProduct[_,_]`: A non commutative product of terms transforming under the Lorentz group

Examples Some examples to clarify the usage and output of `Vertex`:

a) One possible Lorentz structure `Vertex[hh,Ah,Z]` leads to the vertex of scalar and a pseudo scalar Higgs with a Z-boson

```
{{hh[{gt1}], Ah[{gt2}], VZ[{lt3}]},
 {((MA[gt2,1]*MH[gt1,1] - MA[gt2,2]*MH[gt1,2])*(g2*Cos[ThetaW]+g1*Sin[ThetaW]))/2,
               Mom[Ah[{gt2}], lt3] - Mom[hh[{gt1}],lt3]}}
```

The output is divided in two parts. First, the involved particles are given, second, the value of the vertex is given. This second part is again split in two parts: one is the Lorentz independent part and the second part defines the transformation under the Lorentz group.

b) Several possible Lorentz structures `Vertex[bar[Fd],Fd,hh]` is the interaction between d-quarks and a Higgs:

```
{{bar[Fd[{gt1, ct1}]], Fd[{gt2, ct2}], hh[{gt3}]},
 {((-I)*Delta[ct1,ct2]*Delta[gt1,gt2]*MH[gt3,2]*Yd[gt2,gt1])/Sqrt[2],PL},
 {((-I)*Delta[ct1,ct2]*Delta[gt1,gt2]*MH[gt3,2]*Yd[gt1,gt2])/Sqrt[2],PR}}
```

Obviously, there are three parts: one for the involved particles and two for the different Lorentz structures. `PL` and `PR` are the polarization projectors $P_L = \frac{1}{2}(1-\gamma_5), P_R = \frac{1}{2}(1+\gamma_5)$.

c) Changing the considered eigenstates and using Weyl fermions It is also possible to calculate the vertices for Weyl fermions and/or to consider the gauge eigenstates. For instance,

```
Vertex[{fB, FdL, conj[SdL]}, Eigenstates -> GaugeES]
```

gives

```
{{fB, FdL[{gt2, ct2}], conj[SdL[{gt3, ct3}]]},
 {((-I/3)*g1*Delta[ct2, ct3]*Delta[gt2, gt3])/Sqrt[2],1}}
```

d) Using dependences With `Vertex[conj[Se], Se, VP, UseDependences -> True]` g_1 and g_2 are replaced by the electric charge e. This and similar relations can be defined in the parameters file (see sec. 4.2.3).

```
{{conj[Se[{gt1}]], Se[{gt2}], VP[{lt3}]},
{(-I)*e*Delta[gt1,gt2],-Mom[conj[Se[{gt1}]],lt3]+Mom[Se[{gt2}],lt3]}}
```

e) Fixing the generations It is possible to give the indices of the particles already as input

```
Vertex[{hh[{1}], hh[{1}], Ah[{2}], Ah[{2}]}]
```

leads to

```
{{hh[{1}], hh[{1}], Ah[{2}], Ah[{2}]},
 {(-I/4)*(g1^2 + g2^2)*Cos[2*\[Alpha]]*Cos[2*\[Beta]], 1}}
```

Obviously, the given definition of the mixing matrices for the Higgs fields were automatically inserted. If the indices are fixed by a replacement, the definition of the mixing matrix wouldn't be used

```
Vertex[{hh, hh, Ah, Ah}] /. {gt1->1, gt2->1,gt3->2, gt3->2}
```

returns

```
{{hh[{1}], hh[{1}], Ah[{2}], Ah[{gt4}]},
 {(-I/4)*(g1^2 + g2^2)*(conj[ZA[2, 1]]*conj[ZA[gt4, 1]] -
   conj[ZA[2, 2]]*conj[ZA[gt4, 2]])*(conj[ZH[1, 1]]^2 - conj[ZH[1, 2]]^2), 1}}
```

However,

```
Vertex[{hh, hh, Ah, Ah}] /. {gt1->1, gt2->1,gt3->2, gt3->2} /.subAlways
```

leads to the former expression.

f) Effective operators In effective theories also interactions between two fermions and two scalars are possible. As example an effective vertex for a model in which the gluino was integrated out:

```
Vertex[{Fd, Fd, conj[Sd], conj[Sd]}]
```

Returns

```
{{Fd[{gt1, ct1}], Fd[{gt2, ct2}], conj[Sd[{gt3, ct3}]], conj[Sd[{gt4, ct4}]]},
{-(g3^2*(sum[j1, 1, 8, (Lam[j1, ct3, ct2]*Lam[j1, ct4, ct1])/Mass[fG][j1]]*
           ZD[gt3, gt2]*ZD[gt4, gt1] +
         sum[j1, 1, 8, (Lam[j1, ct3, ct1]*Lam[j1, ct4, ct2])/Mass[fG][j1]]*
           ZD[gt3, gt1]*ZD[gt4, gt2])),
    LorentzProduct[PL, PL]}, {0, LorentzProduct[PR, PL]},
{g3^2*(sum[j1, 1, 8, (Lam[j1, ct2, ct3]*Lam[j1, ct4, ct1])/Mass[fG][j1]]*
       ZD[gt3, 3 + gt2]*ZD[gt4, gt1] +
       sum[j1, 1, 8, (Lam[j1, ct2, ct4]*Lam[j1, ct3, ct1])/Mass[fG][j1]]*
       ZD[gt3, gt1]*ZD[gt4, 3 + gt2]),
    LorentzProduct[PL, PR]}, {0, LorentzProduct[PR, PR]},
{0, LorentzProduct[gamma,PL, PL]}, {0, LorentzProduct[gamma, PR, PL]},
{0, LorentzProduct[gamma, PL, PR]}, {0, LorentzProduct[gamma,PR, PR]}}
```

Obviously, SARAH checks the eight possible operators (4 different combination of polarization operators with and without a γ matrix) and returns the result for each operator.

3.5 Renormalization Group Equations

SARAH calculates the renormalization group equations (RGEs) for the parameters of the superpotential, the soft-breaking terms and the gauge couplings at one and two loop level. This is done by using the generic formulas of [20].

The calculation is started via

```
CalcRGEs[Options]
```

Options The different options are

a) `TwoLoop`, Value: `True` or `False`, Default: `True`
If also the two loop RGEs should be calculated.

b) `ReadLists`, Value: `True` or `False`, Default: `False`
If the RGEs have already be calculated, the results are saved in the output directory. The RGEs can be read from these files instead of doing the complete calculation again.

c) `VariableGenerations`, Value: List of particles, Default: `{}`
Some theories contain heavy superfields which should be integrated out above the SUSY scale. Therefore, it is possible to calculate the RGEs assuming the number of generations of specific superfields as free variable to make the dependence on these fields obvious. The new variable is named `NumberGenertions[X]`, where `X` is the name of the superfield.

d) `NoMatrixMultiplication`, values: `True` or `False`, Default: `False`
Normally, the β-functions are simplified by writing the sums over generation indices as matrix multiplication. This can be switched off using this option.

The β-functions will be stored in the following arrays:

a) `Gij`: Anomalous dimensions of all chiral superfields

b) `BetaYijk`: Trilinear superpotential parameters

c) `BetaMuij`: Bilinear superpotential parameters

d) `BetaLi`: Linear superpotential parameters

e) `BetaTijk`: Trilinear soft-breaking parameters

f) `BetaBij`: Bilinear soft-breaking parameters

g) `BetaSLi`: Linear soft-breaking parameters

h) `Betam2ij`: Scalar squared masses

i) `BetaMi`: Gaugino masses

j) `BetaGauge`: Gauge couplings

k) `BetaVEVs`: VEVs

These arrays are also saved in the directory

```
../\SARAH/Output/"ModelName"/RGE
```

All entries of this arrays are three dimensional: The first entry is the name of the parameter, the second the one-loop β-function and the third one the two loop β-function.

GUT normalization The gauge couplings of $U(1)$ gauge groups are often normalized at the GUT scale. Therefore, it is possible to define for each gauge coupling the GUT-normalization by the corresponding entry in the parameters file. See sec. 4.2.3 for more information. Generally, the results contain sums over the generation indices of the particles in the loop. SARAH always tries to write them as matrix multiplications, in order to shorten the expressions. Therefore, new symbols are introduced:

a) `MatMul[A,B,C,...][i,j]`: $(ABC\dots)_{i,j}$. Matrix multiplication, also used for vector-matrix and vector-vector multiplication.

b) `trace[A,B,C,...]`: $\mathrm{Tr}(ABC\dots)$. Trace of a matrix or product of matrices.

c) `Adj[M]`: $M^\dagger$. Adjoint of a matrix

d) `Tp[M]`: M^T. Transposed of a matrix

Remarks Some remarks about the output:

a) To differ between generation and other indices during the calculation, `Kronecker[i,j]` is used for generation indices instead of `Delta[i,j]`.

b) The results for the scalar masses are simplified by using abbreviations for often appearing traces. The definition of the traces are saved in the array `TraceAbbr`.

c) If the model contains parameters with three indices, matrix multiplication is automatically switched off.

Examples

a) **β-function of Yukawa coupling** The Yukawa couplings of the MSSM are saved in `BetaYijk`. The first entry consists of

```
BetaYijk[[1,1]]:  Ye[i1,i2] ,
```

i.e. this entry contains the β-functions for the electron Yukawa coupling. The corresponding one-loop β-function is

```
BetaYijk[[1,2]]:
(-9*g1^2*Ye[i1,i2])/5-3*g2^2*Ye[i1,i2]+3*trace[Yd,Adj[Yd]]*Ye[i1,i2]+
  trace[Ye,Adj[Ye]]*Ye[i1, i2]+3*MatMul[Ye,Adj[Ye],Ye][i1, i2]
```

The two-loop β-function is saved in `BetaYijk[[1,3]]` but we skip it here because of its length.

b) **β-function of soft-breaking masses and abbreviations for traces** The soft-breaking mass of the selectron is the first entry of `Betam2ij`

```
 Betam2ij[[1,1]]:           me2[i1,i2]
```

and the one-loop β-function is saved in `Betam2ij[[1,2]]`:

```
(-24*g1^2*MassB*conj[MassB]+10*g1^2*Tr1[1])*Kronecker[i1,i2]/5 +
 4*mHd2*MatMul[Ye,Adj[Ye]][i1,i2]+4*MatMul[T[Ye],Adj[T[Ye]]][i1,i2] +
  2*MatMul[me2,Ye,Adj[Ye]][i1,i2]+4*MatMul[Ye, ml2, Adj[Ye]][i1,i2] +
  2*MatMul[Ye,Adj[Ye],me2][i1,i2]
```

The definition of the element Tr1[1] is saved in TraceAbbr[[1,1]]:

```
{Tr1[1], -mHd2 + mHu2 + trace[md2] + trace[me2] - trace[ml2] +
          trace[mq2] - 2*trace[mu2]}
```

c) Number of generations as variable: With

```
CalcRGEs[VariableGenerations -> {q}]
```

the number of generations of the left-quark superfield is handled as variable. Therefore, the one-loop β-function of the hypercharge couplings reads

```
(63*g1^3)/10 + (g1^3*NumberGenerations[q])/10
```

d) No matrix multiplication Using matrix multiplication can be switched off by

```
CalcRGEs[NoMatrixMultiplication -> True]
```

The one-loop β-function for the electron Yukawa coupling is now written as

```
  sum[j2,1,3,sum[j1,1,3,conj[Yd[j2,j1]]*Yu[i1,j1]]*Yd[j2,i2]] +
2*sum[j2,1,3,sum[j1,1,3,conj[Yu[j1,j2]]*Yu[j1,i2]]*Yu[i1,j2]] +
  sum[j2,1,3,sum[j1,1,3,conj[Yu[j2,j1]]*Yu[i1,j1]]*Yu[j2,i2]] +
(3*sum[j2,1,3,sum[j1,1,3,conj[Yu[j1,j2]]*Yu[j1,j2]]]*Yu[i1,i2])/2 +
(3*sum[j2,1,3,sum[j1,1,3,conj[Yu[j2,j1]]*Yu[j2,j1]]]*Yu[i1,i2])/2 -
 (13*g1^2*Yu[i1,i2])/15-3*g2^2*Yu[i1,i2]-(16*g3^2*Yu[i1,i2])/3
```

3.6 Loop Corrections

SARAH calculates the analytical expressions for the one-loop corrections to the tadpoles and the self energy of all particles. These calculations are performed in $\overline{DR}$-scheme and in the 't Hooft gauge. To command to start the calculation is

```
CalcLoopCorrections[Eigenstates];
```

As usual, Eigenstates can in the case of the MSSM either be GaugeES for the gauge eigenstates or EWSB for the eigenstates after EWSB. If the vertices for the given set of eigenstates were not calculated before, this is done before the calculation of the loop contributions begins.

Conventions Using the conventions of [21], the results will contain the Passarino Veltman integrals listed in app. E.2. The involved couplings are abbreviated by

a) Cp[p1,p2,p3] and Cp[p1,p2,p3,p4] for non-chiral, three and four point interactions involving the particles p1 - p4.

b) Cp[p1,p2,p3][PL] and Cp[p1,p2,p3][PR] for chiral, three-point interactions involving the fields p1 - p3.

The self energies can be used for calculating the radiative corrections to masses and mass matrices, respectively. We have summarized the needed formulas for this purpose in app. E.2.3. For calculating the loop corrections to a mass matrix, it is convenient to use unrotated, external fields, while the fields in the loop are rotated. Therefore, SARAH adds to the symbols of the external particle in the interaction an U for 'unrotated', e.g. Sd $\rightarrow$ USd. The mixing matrix associated to this field in the vertex has to be replaced by the identity matrix when calculating the correction to the mass matrix.

Results The results for the loop corrections are saved in two different ways. First as list containing the different loop contribution for each particle. Every entry reads

```
{Particles, Vertices, Type,Charge Factor, Symmetry Factor}
```

and includes the following information

a) `Particles`: The particles in the loop.

b) `Vertices`: The needed Vertex for the correction is given.

c) `Charge Factor`: If several gauge charges of one particle are allowed in the loop, this factor will be unequal to one. In the case of the MSSM, only the a factor of 3 can appear because of the different colors.

d) `Symmetry Factor`: If the particles in the loop indistinguishable, the weight of the contribution is only half of the case of distinguishable particles. If two different charge flows are possible in the loop, the weight of the diagram is doubled, e.g. loop with charged Higgs and W-boson. The absolute value of the factor depends on the type of the diagram.

The information about the loop correction are also saved in the directory

```
../\SARAH/Output/"ModelName"/"Name of Eigenstates"/Loop
```

One Loop Tadpoles The complete results as sums of the different contributions are saved in the two dimensional array

```
Tadpoles1LoopSums["Name of Eigenstates"]
```

The first column gives the name of the corresponding VEV, the second entry the one-loop correction. A list of the different contributions, including symmetry and charge factors, is

```
Tadpoles1LoopList["Name of Eigenstates"];
```

One Loop Self Energies The results are saved in the following two dimensional array

```
SelfEnergy1LoopSum["Name of Eigenstates"]
```

The first column gives the name of the particle, the entry in the second column depends on the type of the field

a) Scalars: one-loop self energy $\Pi(p^2)$

b) Fermions: one-loop self energies for the different polarizations ($\Sigma^L(p^2)$,$\Sigma^R(p^2)$, $\Sigma^S(p^2)$)

c) Vector bosons: one-loop, transversal self energy $\Pi^T(p^2)$

Also a list with the different contributions does exist:

```
SelfEnergy1LoopList["Name Of States"]
```

Examples

a) One-loop tadpoles The correction of the tadpoles due to a chargino loop is saved in

```
Tadpoles1LoopList[EWSB][[1]];
```

and reads

```
{bar[Cha],Cp[Uhh[{gO1}],bar[Cha[{gI1}]],Cha[{gI1}]],FFS,1,1/2}
```

The meaning of the different entries is: (i) a chargino (`Cha`) is in the loop, (ii) the vertex with an external, unrotated Higgs (`Uhh`) with generation index `gO1` and two charginos with index `gI1` is needed, (iii) the generic type of the diagram is `FFS`, (iv) the charge factor is 1, (v) the diagram is weighted by a factor $\frac{1}{2}$ with respect to the generic expression (see app. E.2).
The corresponding term in `Tadpoles1LoopSum[EWSB]` is

```
4*sum[gI1,1,2, A0[Mass[bar[Cha[{gI1}]]]^2]*
   Cp[phid,bar[Cha[{gI1}]],Cha[{gI1}]]*Mass[Cha[{gI1}]]]
```

b) One-loop self-energies

(a) The correction to the down squark matrix due to a four point interaction with a pseudo scalar Higgs is saved in `SelfEnergy1LoopList[EWSB][[1, 12]]` and reads

```
{Ah,Cp[conj[USd[{gO1}]],USd[{gO2}],Ah[{gI1}],Ah[{gI1}]],SSSS,1,1/2}
```

This has the same meaning as the term

```
-sum[gI1,1,2,A0[Mass[Ah[{gI1}]]^2]*
   Cp[conj[USd[{gO1}]],USd[{gO2}],Ah[{gI1}],Ah[{gI1}]]]/2
```

in `SelfEnergy1LoopSum[EWSB]`.

(b) Corrections to the Z boson are saved in `SelfEnergy1LoopList[EWSB][[15]]`. An arbitrary entry looks like

```
{bar[Fd], Fd, Cp[VZ, bar[Fd[{gI1}]], Fd[{gI2}]], FFV, 3, 1/2}
```

and corresponds to

```
(3*sum[gI1, 1, 3, sum[gI2, 1, 3,
     H0[p^2, Mass[bar[Fd[{gI1}]]]^2, Mass[Fd[{gI2}]]^2]*
 (conj[Cp[VZ,bar[Fd[{gI1}]],Fd[{gI2}]][PL]]*
     Cp[VZ,bar[Fd[{gI1}]],Fd[{gI2}]][PL] +
  conj[Cp[VZ,bar[Fd[{gI1}]],Fd[{gI2}]][PR]]*
     Cp[VZ,bar[Fd[{gI1}]],Fd[{gI2}]][PR]) +
 2*B0[p^2,Mass[bar[Fd[{gI1}]]]^2,Mass[Fd[{gI2}]]^2]*
     Mass[bar[Fd[{gI1}]]]*Mass[Fd[{gI2}]]*
  Re[Cp[VZ,bar[Fd[{gI1}]],Fd[{gI2}]][PL]*
     Cp[VZ,bar[Fd[{gI1}]],Fd[{gI2}]][PR])]])/2
```

in `SelfEnergy1LoopListSum[EWSB]`.

3.7 Output for LaTeXand diagram calculators

With

```
ModelOutput[Eigenstates,Options]
```

LaTeX-files and model files for `FeynArts` and `CompHep`/`CalcHep` can be generated. Here, `Eigenstates` specifies the eigenstates which should be used, e.g. `GaugeES` or `EWSB`. The options are the following

a) `WriteTeX`, values: `True` or `False`, Default: `False`
If a LaTeXfile containing all information about the model should be written.

b) `WriteFeynArts`, values: `True` or `False`, Default: `False`
If a model file for `FeynArts` should be written.

c) `WriteCHep`, values: `True` or `False`, Default: `False`
If a model file for `CompHep`/`CalcHep` should be written.

d) `effectiveOperators`, Values: `True` or `False`, Default: `False`
If also higher dimensional operators should be calculated. By default, this concerns only four point interactions.

e) `SixParticleInteractions`, Values: `True` or `False`, Default: `False`
If also the six-point interactions should be calculated.

f) `FeynmanDiagrams`, Values: `True` or `False`, Default: `True`
If Feynman diagrams for each vertex should be drawn in the LaTeXfile.

g) `ReadLists`, Values: `True` or `False`, Default: `False`
If the results of former calculations should be used to save time.

h) `IncludeRGEs`, Values: `True` or `False`, Default: `True`
If the RGEs should be calculated.

i) `TwoLoopRGEs`, Values: `True` or `False`, Default: `True`
If the two loop RGEs should be calculated. (`IncludeRGEs` must be set to `True`)

j) `IncludeLoopCorrections`, Values: `True` or `False`, Default: `True`
If the one-loop corrections to the self-energy and the tadpoles should be calculated.

The generated output will be saved in

```
../\SARAH/Output/"ModelName"/"Name of Eigenstates"/TeX/"
../\SARAH/Output/"ModelName"/"Name of Eigenstates"/\FeynArts/"
../\SARAH/Output/"ModelName"/"Name of Eigenstates"/CHep/"
```

More details about the output is given in chapter sec. 6.

Chapter 4

The model files

All information of the different models are saved in three different files which have to be in one directory

`.../\SARAH/Models/"ModelName"/`

> The directory name is equal to the name of the model and must contain a model file with the same name!

The three files are: one model file with the same name as the directory (`ModelName.m`), a file containing additional information about the particles of the model (`particles.m`) and a file containing additional information about the parameters of the model (`parameters.m`). Only the first file is really necessary for calculating the Lagrangian and to get a first impression of a model. However, for defining properties of parameters and particles and for producing an appropriate output the other two files are needed.
In addition, it is possible to include spectrum files in the model directory. These can be read without the necessity to give the complete path (see also 8.1).

4.1 The model file

The model file is the heart of `SARAH`: the complete model is specified by the model file. Thus, we will explain the general structure of the model file here, and have a look at the different functions and its physical meaning in detail in the next chapter.

4.1.1 Description

The model file contains the following parts: First the gauge structure and the particle content are given, and the matter interactions are defined by the superpotential. These are general information needed for all eigenstates of the model and must always be apparent.
This part is followed by the definition of the names for all eigenstates (`NameOfStates`). For these eigenstates can afterwards several properties be defined using the corresponding `DEFINITION` statement: decomposition of scalars in scalar, pseudo scalar and VEV (`DEFINITION[States][VEVs]`), rotations in the matter (`DEFINITION[States][MatterSector]`) and gauge sector (`DEFINITION[States][GaugeSector]`) and the corresponding gauge fixing terms (`DEFINITION[States][GaugeFixing]`). New couplings can be added and existing couplings can be changed by hand (`DEFINITION[States][Additional]`).
Afterwards, the particles are states, which should be integrated out or deleted. At the end, the Dirac spinors have to be built out of Weyl spinors, a spectrum file can be defined and a choice for automatically output can be made.

4.1.2 Schematic Structure

The model file is structured as follows

General information

a) Gauge structure of the model given by the vector superfields, see sec. 5.1.1

```
Gauge[[1]] = {...
```

b) Matter content given by the chiral superfields see sec. 5.1.2

```
Fields[[1]] = { ...
```

c) Superpotential, see sec. 5.2

```
SuperPotential { ...
```

Eigenstates

Names for the different eigenstates, see sec. 5.3

```
NameOfStates = { ...
```

Definition of properties for the different eigenstates

a) Vacuum expectation values, see sec. 5.3.1.1

```
DEFINITION["Name of Eigenstates"][VEVs]= { ...
```

b) Rotations in the gauge sector, see sec. 5.3.3

```
DEFINITION["Name of Eigenstates"][GaugeSector]= { ...
```

c) Expansion of flavors, see sec. 5.3.2

```
DEFINITION["Name of Eigenstates"][Flavors]= { ...
```

d) Rotations in the matter sector, see sec. 5.3.4

```
DEFINITION["Name of Eigenstates"][MatterSector]= { ...
```

e) Gauge fixing terms, see sec. B

```
DEFINITION["Name of Eigenstates"][GaugeFixing]= { ...
```

f) Additional couplings or redefinition of existing couplings, see sec. 5.3.6.2

```
DEFINITION["Name of Eigenstates"][Additional]= { ...
```

Additional, general Information

a) Definition of Dirac spinors, see sec. 5.5

```
dirac[[1]] = { ...
```

b) Integrating out or deleting particles, see sec. 5.4

```
IntegrateOut[[1]] = { ...
DeleteParticles[[1]] = { ...
```

c) Automatized output, see sec. 6.5

```
makeOutput = { ...
```

d) Assigning a spectrum file, see sec. 8.1

```
SpectrumFile = ...
```

4.2 Parameter file

4.2.1 General

The information of the parameter file are needed for some in- and output routines. Also, it is possible to define simplifying assumptions for the parameters. The parameter file consists of a list called `ParameterDefinition`. This is an array with two columns: the first column gives the name of the parameter, the second column defines the properties of the parameter. These properties can be:

a) `Description`: Value: a string
A string for defining the parameter, see sec. 4.4

b) `Real`, Value: `True` or `False`, Default: `False`
Defines, if a parameter is always assumed to be real.

c) `Form`, Value: `None`, `Diagonal` or `Scalar`, Default: `None`
For a explanation of the different options see sec. 4.2.2.

d) `LaTeX`, Value: `None` or LaTeXname
The name of the parameter used in the LaTeXoutput. Standard LaTeXlanguage should be used (\ has to be replaced by \\).

e) `Dependence`, Value: `None` or function
The parameter is always replaced by this definition, see sec. 4.2.3

f) `DependenceOptional`, Value: `None` or function
It can be chosen during the work, if the parameter is replaced by the made definition, see sec. 4.2.3

g) `DependenceNum`, Value: `None` or function
This definition is used in numerical calculations, see sec. 4.2.3

h) `MatrixProduct`, Value: `None` or list of two matrices
The parameter is defined as a product of two matrices, see sec. 4.2.3

i) `LesHouches`, Value: `None` or position in LesHouches input file
The numerical value of the parameter can be given by a LesHouches file, see sec. 4.2.3

j) `Value`, Value: `None` or number
A numerical value for the parameter can be chosen, see sec. 4.2.3

4.2.2 Simplifying assumptions

SARAH normally handles parameters in the most general way: most parameters assumed to be complex and all tensors can have off diagonal values. This can be changed by certain statements in the parameter file. First, it is possible to define a parameter as real by setting

```
Real -> True
```

The gauge couplings are by default assumed to be real.
The degrees of freedom for a tensor valued parameter T can be reduced by using the `Form` statement with the following options

a) `Diagonal`: only diagonal entries are assumed to be unequal from zero: $T_{ij} \to \delta_{ij} T_{ij}$

b) `Scalar` the tensor is replaced by a scalar: $T_{ij} \to T$

Furthermore, some symmetries can be defined for a parameter by using the option `Symmetry`

a) `Symmetric`: the tensor is assumed to be symmetric: $T_{ij} = T_{ji}$ for $i > j$

b) `AntiSymmetric`: the tensor is assumed to be anti-symmetric: $T_{ij} = -T_{ji}$ for $i > j$

c) `Hermitian`: the tensor is assumed to be hermitian: $T_{ij} = T^*_{ji}$ for $i > j$

Examples

a) CP and Flavor conserving Yukawa matrices are defined by

```
{Yu,  { Real -> True,
        Form -> Diagonal}}
```

b) All generations of gauginos have the same mass, so the parameter can in most cases expressed by a scalar instead of a matrix:

```
{MassG,  { Real -> True,
           Form -> Scalar}}
```

4.2.3 Defining values and dependences for parameters

4.2.3.1 Dependences

There are different possibilities to define dependences between parameters by using the `Dependence` statements. The difference between the three statements is the time at which the relations are used

a) `Dependence`: The relations are always used. The corresponding substitutions are saved in `subAlways`.

b) `DependenceOptional`: The relations are only used if the option `UseDependence` is set to `True`, e.g. when calculating vertices. The substitutions are saved in `subDependences`.

c) `DependenceNum`: The dependences are only used when a numerical value for the parameter is calculated. The substitutions are saved in `subNum`.

The indices of vectors or tensors are implicitly assumed to be `index1`, `index2`, ... This in combination with `sum[index,start,final]` can be used in the following way

```
{X, {Dependences -> sum[n1,1,3] sum[n2,1,3] A[index1,n1] Y[n1,n2] B[n2,index2]}}
```

is interpreted as

$$X_{i_1 i_2} \to \sum_{n_1=1}^{3} \sum_{n_2=1}^{3} A_{i_1 n_1} Y_{n_1 n_2} B_{n_2 i_2} \tag{4.1}$$

Parameters can also defined as matrix product of other matrices by using `MatrixProduct`. The argument must be a list consisting of two matrices of same dimension:

```
{X, {MatrixProduct-> {A,B} }}
```

Using this definition, every appearance of a matrix product of $A^\dagger B$ is replaced by X and $B^\dagger A$ by $X^\dagger$.

Examples

a) One dependence, which might be used always, is the parametrization of a mixing matrix by a mixing angle: the mixing of the charged Higgs (`ZP`) in the MSSM can be parametrized by a mixing angle β. This is defined in `SARAH` via

```
{ZP,  { Dependence -> {{-Cos[\[Beta]],Sin[\[Beta]]},
                       {Sin[\[Beta]],Cos[\[Beta]]}}
```

b) The relation between the gauge couplings g_1 and g_2 and the electric charge e is an example for an optional dependence. This relation is defined by

```
{g1,  {DependenceOptional -> e/Cos[ThetaW]}}
```

Now, the result for vertices can be expressed in terms of the electric charge by

```
Vertex[List of Particles, UseDependences -> True]
```

c) A relation, which might only be used for numerical calculations, is the relation between the Weinberg angle and the gauge couplings:

```
{ThetaW, {DependenceNum ->  ArcCos[g2/Sqrt[g1^2+g2^2]]}
```

d) The CKM matrix is defined as the product of two rotation matrices:

```
{CKM, {MatrixProduct -> {Vd,Vu} }}
```

4.2.3.2 Numerical values

If the considered parameter does not depend on other parameters, there are two ways to assign a numerical value to this parameter:

a) `Value`: Adds directly a numerical value to the parameter definition

b) `LesHouches`: Defines the position of the numerical value for the parameter in a LesHouches spectrum file (see also [19]). The statement has to has the following form:

(a) If the dimension of the parameters is known by SARAH, it is sufficient to give just the name of the block, e.g.

```
{Yu, { LesHouches -> Yu } };
```

(b) If the dimension of the parameters is not known by SARAH, a list of entries hat to be given, e.g.

```
{Ae, {LesHouches -> {{{1,1}, {Ae,1,1}},
                     {{2,2}, {Ae,2,2}},
                     {{3,3}, {Ae,3,3}}}};
```

(c) If the block appears several times in the LesHouches file, it is possible to give the number of appearance as optional argument:

```
{g3,  {LesHouches -> {gauge,3}}};
```

This reads the first block which are normally the GUT-scale values. However,

```
{g3,  {LesHouches -> {{gauge,3},2}}};
```

reads the second block which are the values at the SUSY-scale.

Example

a) The numerical values of the CKM matrix are known, and can be defined as

```
{CKM,  {Value -> {{{1,1}, 0.97383},
                  {{1,2}, 0.2272},
                  {{1,3}, 0.00396},
                  {{2,1}, 0.2271},
                  {{2,2}, 0.97296},
                  {{2,3}, 0.04221},
                  {{3,1}, 0.00814},
                  {{3,2}, 0.04161},
                  {{3,3}, 0.999100}}
        }}
```

b) The values for the soft-breaking parameters are normally calculated by using a spectrum calculator and written in a LesHouches file. To read this file with SARAH, it is possible to define the position for the values in the spectrum file with

```
{md,  { LesHouches -> {{{1,1}, {MSOFT,47}},
                       {{2,2}, {MSOFT,48}},
                       {{3,3}, {MSOFT,49}}}
      }}
```

4.3 Particles File

The particle file contains information about the fields of the model, which are needed for the out- and input.

a) `Description`: A string for defining the particle

b) `RParity`: The R-Parity of a particle: +1 or -1. If not defined, +1 is used.

c) `PDG`: The PDG number. Needed, if the written model files should be readable by event generators or if masses are given by a LesHouches file.

d) `Width`: The width of the particle. If not defined, 0 is used.

e) `Mass`: The options of defining the mass of a particle are:

(a) Numerical Value: A numerical value for the mass of the particle is given.

(b) `Automatic`: `SARAH` derives the tree level expression for the mass from the Lagrangian and calculates the value by using the values of the other parameters.

(c) `LesHouches: SARAH` reads the mass from a LesHouches file.

f) `LaTeX`: The name of the particle in LaTeXfiles in standard LaTeXlanguage. If not defined, the Mathematica `InputForm` of the particle name is used.

g) `\FeynArtsNr`: The number of the particle in a `FeynArts` model file. If not defined, the number will be generated automatically.

h) `Output`: A short form the particle name consisting of two letter with no no-standard signs. Needed, to make sure that all programs outside `SARAH` can read the name correctly. If not defined, the Mathematica `InputForm` is used and potentially truncated.

This information must be given for all eigenstates in arrays named

```
ParticleDefinition["Name of Eigenstates"]
```

Only for the LaTeXoutput also the names of the Weyl spinors and intermediate states (like scalar and pseudo scalar components of Higgs) should be given to improve the layout of the produced pdf file.

Example For the eigenstates after EWSB, an entry might look like

```
ParticleDefinitions[EWSB} = {
   ...
   {Sd ,  {  Description -> "Down Squark",
             RParity -> -1,
             PDG ->  {1000001,2000001,1000003,2000003,1000005,2000005},
             Width -> 0,
             Mass -> Automatic,
             \FeynArtsNr -> 14,
             LaTeX -> "\\tilde{d}",
             OutputName -> "dm" }},
   ... }
```

4.4 Global definitions

It is also possible to define global properties for parameters or particles which are present in more than one model file. These properties are afterwards used for all models. The global definitions are saved in the files `particles.m` and `parametes.m` directly in the main model directory. For each parameter or particle, an entry like

```
{{        Descriptions -> "Down Squark",
          RParity -> -1,
          PDG ->  {1000002,2000002,1000004,2000004,1000006,2000006},
          Width -> Automatic,
          Mass -> Automatic,
          FeynArtsNr -> 13,
          LaTeX -> "\\tilde{u}",
          OutputName -> "um" }},
```

can be added. In particular, the entry `Description` is important. This should be an unique identifier for each particle or parameter. This identifier can later on be used in the different files of the different models, e.g.

```
{Su ,  {  Descriptions -> "Down Squark"}},
```

Of course, it is also possible to overwrite some of the global definitions by defining them locally, too. For instance, to use `u` instead of `um` as output name in a specific model, the entry should be changed to

```
{Su ,  {  Descriptions -> "Down Squark",
          OutputName -> "u" }},
```

in the corresponding particle file of the model.

Chapter 5

Definition of models

5.1 Particle content

5.1.1 Vector superfields

The vector superfields are defined by the array `Gauge`. An entry reads

```
Gauge[[i]]={Superfield Name, Dimension, Name of Gauge Group, Coupling, Expand};
```

The different parts have the following meaning:

a) `Superfield name`:
This is the name for the vector superfield and also the basis of the names for vector bosons and gauginos as explained in sec. 3.1

b) `Dimension`:
This defines the dimension of the $SU(N)$ gauge group: `U[1]` for an abelian gauge group or `SU[N]` with integer N for a non-abelian gauge group.

c) `Name of Gauge Group`:
This is the name of the gauge group, e.g. hypercharge, color or left. This choice is import because all matter particles charged under a non-abelian gauge group carry an corresponding index. The name of the index consists of the first three letter of the name plus a number. Hence, it must be taken care that the first three letters of different gauge group names are not identical

d) `Coupling`: The name of the coupling constant, e.g. `g1`

e) `Expand`: Values can be `True` or `False`. If it is set to `True`, all sums over the corresponding indices are evaluated during the calculation of the Lagrangian. This is normally done non-abelian gauge groups which get broken like the $SU(2)_L$ in the MSSM.

`SARAH` adds for every vector superfield a soft-breaking gaugino mass

```
Mass<>"Superfield Name"
```

Example: Standard model color group

```
Gauge[[3]] = {G,  SU[3],  color,  g3, False};
```

The consequence of this entry is

a) Gluon and gluino are named `VG` respectively `fG`

b) The $SU(3)$ generators, the Gell-Mann matrices, are used

c) The color index is abbreviated `colX` (for `X` = 1,2, ...)

d) The strong coupling constant is named `g3`

e) The sums over the color indices are not evaluated

5.1.2 Chiral Superfields

The matter fields are arranged in chiral superfields.

```
Field[[i]] = {Components, Generations, Superfield Name, Transformation 1, Transformation 2... };
```

a) `Components`: The basis of the name for the components. Two cases are possible:

(a) The field transforms only trivially under the gauge groups with expanded indices. In this case, the entry is one dimensional.

(b) The field transforms non-trivially under gauge groups with expanded indices. In this case, the entry is a vector or higher dimensional tensor fitting to the dimension of the field.

b) `Generations`: The number of generations

c) `Superfield Name`: The name for the superfield

d) `Transformation 1`, `Transformation 2`,..: Transformation under the different gauge groups defined before. For $U(1)$ this is the charge, for non-abelian gauge groups the dimensions is given as integer respectively negative integer. The dimension D of an irreducible representation is not necessarily unique. Therefore, to make sure, `SARAH` uses the demanded representation, also the corresponding Dynkin labels have to be added.

`SARAH` adds automatically for all chiral superfields soft-breaking squared masses named

```
m <> "Name of Superfield" <> 2
```

In addition, there are mixed soft-breaking parameters added which are allowed by gauge invariance and R-parity. The last check can be disabled by setting

```
RParityConservation = False;
```

in the model file.

Examples

a) Fields with expanded indices The definition of the left quark superfield in the MSSM is

```
Field[[1]] = {{uL,  dL},  3, q,   1/6, 2, 3};
```

The consequence of this definition is

(a) Left up-squarks and quarks are called `SuL` / `FuL`
(b) Left down-squarks and quarks are called `SdL` / `FdL`
(c) There are three generations
(d) The superfield is named `q`
(e) The soft-breaking mass is named `mq2`
(f) The hypercharge is $\frac{1}{6}$
(g) The superfield transforms as **2** under $SU(2)$
(h) The superfield transforms as **3** under $SU(3)$

b) Fields with no expanded indices The right down-quark superfield is defined in the MSSM as

```
Field[[3]] = {{conj[dR]}, 3, d, 1/3, 1, -3};
```

The meaning is

(a) The right squarks and quarks are called `SdR` and `FdR`
(b) There are three generations
(c) The Superfield name is `d`
(d) The soft-breaking mass is named `md2`
(e) The hypercharge is $\frac{1}{3}$
(f) It does not transform under $SU(2)$
(g) It does transform as $\bar{\mathbf{3}}$ under $SU(3)$

c) Specification of representation Since the **10** under $SU(5)$ is not unique, it is necessary to add the appropriate Dynkin labels, i.e.

```
Field[[1]] = {Ten, 1, t, {10,{0,1,0,0}},...};
```

or

```
Field[[1]] = {Ten, 1, t, {10,{0,0,1,0}},...};
```

d) Mixed soft-breaking terms In models which contain fields with the same quantum numbers and same R-parity mixed soft-breaking terms are added. For instance, in models with heavy squarks

```
Field[[3]] = {{conj[dR]}, 3, d, 1/3, 1, -3};
...
Field[[10]] = {{conj[dRH]}, 3, DH, 1/3, 1, -3};
```

the term of the form

```
mdDH (conj[SdR] SdRH + SdR conj[SdRH])
```

is automatically added. For the MSSM with R-parity violation, also the term

```
mlHd (conj[Sl] SHd + Sl conj[SHd])
```

is created.

5.2 Superpotential

The definition of the superpotential is straight forward

```
SuperPotential = {{{Coefficient,Parameter,(Contraction)},
                      {Particle 1, Particle 2, (Particle 3)} }, ...}
```

Each term of the superpotential is defined due to two list:

a) The first list is two dimensional and defines a numerical coefficient (`Coefficient`) and the name of the parameter (`Parameter`)

b) The second list consists of one to three particles

c) The contraction of the indices can be given optionally.

Contraction of indices The indices of the involved particles are automatically contracted by `SARAH`. Sometimes, there are more possibilities to contract all indices. Therefore, it is also possible to fix the contraction of each term. The contraction used by `SARAH` can be seen by

```
ShowSuperpotentialContractions;
```

Properties of couplings and soft-breaking terms If the particles involved in the different interactions have more than one generation, the couplings are in general complex tensors carrying up to three generation indices. Assumptions like diagonality or no CP violation can added by using the parameter file (see sec. 4.2).
The corresponding soft-breaking term to each superfield coupling is automatically added to the Lagrangian. The soft-breaking couplings carry the same indices as the superpotential coupling. They are named as

a) Trilinear terms: `T["Name of Coupling"]`

b) Bilinear terms: `B["Name of Coupling"]`

c) Linear terms: `L["Name of Coupling"]`

Simplifying assumptions for the soft-breaking terms can be made independently of the assumptions for the superpotential parameters in the parameter file.

Example The term involving the up Yukawa coupling is

$$Y^u_{n_1,n_2}\hat{q}^i_{\alpha,n_1}\epsilon^{ij}\hat{H}^{\hat{j}}_u u_{\beta,n_2}\delta_{\alpha,\beta} \tag{5.1}$$

This can be defined in SARAH with

```
{{1, Yu},{q,Hu,u}}
```

The explicit contraction of the indices would read as

```
{{1, Yu,Delta[col1,col3] Eps[lef1,lef2] },{q,Hu,u}}
```

There is a soft-breaking term automatically added to Lagrangian, which has the same meaning as

$$T(Y^u)_{n_1,n_2}\tilde{q}^i_{\alpha,n_1}\epsilon^{ij}H^j_u\tilde{u}^*_{\beta,n_2}\delta_{\alpha,\beta} \tag{5.2}$$

5.3 Properties of different eigenstates

For defining the properties of the different sets of eigenstates, the DEFINITION statement is used:

```
DEFINITION["Name of States"]["Property"]= {...};
```

The possible properties are VEVs, GaugeSector, MatterSector, GaugeFixing, Phase, Flavors and Additional.
But, before defining this properties, the names for all eigenstates must be fixed in the correct order. This is done due to

```
NameOfStates={List of Name};
```

This list can, in principle, be arbitrary long. Common entries are, e.g.

```
NameOfStates={GaugeES, EWSB}
NameOfStates={GaugeES, SCKM, EWSB}
```

5.3.1 Vacuum expectation values

5.3.1.1 Introduction

The particles responsible for breaking a gauge symmetry receive a VEV. After the symmetry breaking, these particles are parametrized by a scalar ϕ and a pseudo scalar σ part and the VEV v:

$$S = \frac{1}{\sqrt{2}}(\phi_S + i\sigma_S + v_S) \tag{5.3}$$

5.3.1.2 Implementation in SARAH

This is in SARAH done by

```
DEFINITION["Name of Eigenstate"][VEVs] =
{Particle Name, {{VEV, Coefficient 1},
      {Pseudoscalar, Coefficient 2},{Scalar, Coefficient 3}};
```

a) Name: The name of the particle receiving a VEV

b) `VEV`: Name of the VEV

c) `Scalar`: Name of the scalar component

d) `Pseudoscalar`: Name of the pseudo scalar component

e) `Coefficient 1,2,3`: The different (numerical) coefficients.

All indices carried by the particle receiving the VEV are automatically added to the scalar and pseudo scalar part. The scalar, pseudo scalar and the VEV are handled as real parameters in `SARAH`.

Example In the MSSM, the Higgs H_d^0 gets a VEV v_u and is therefore parametrized by

$$H_u^0 = \frac{1}{\sqrt{2}} (v_u + i\sigma_u + \phi_u) \tag{5.4}$$

This is done in `SARAH` by using

```
DEFINITION[EWSB][VEVs]=
{{SHd0, {vd, 1/Sqrt[2]}, {sigmad, \[ImaginaryI]/Sqrt[2]},{phid,1/Sqrt[2]}}};
```

5.3.2 Decomposition of Flavors

5.3.2.1 Introduction

If a model without flavor violation is considered, it might be demanded to give each generation of a family a own name, e.g. use different symbols for electron, muon and tau.

5.3.2.2 Implementation in `SARAH`

The expansion of flavors is done by

```
DEFINITION[EWSB][Flavors]= { ...
 {Field, {Name 1, Name 2, ... }},
 ... }
```

There must be as many names as generations of `Field` exist.

Example The down-type quarks can be expanded as follows:

```
DEFINITION[EWSB][Flavors]= {
 {FdOL, {FdL,FsL,FbL}},
 {FdOR, {FdR,FsR,FbR}} }
```

5.3.3 Mixings in the Gauge Sector

5.3.3.1 Introduction

After breaking a gauge symmetry, the vector bosons mix among each other as well as the gauginos do. In general, the different generations of a vector boson or gaugino rotate to different mass eigenstates. Hence, it is not possible to use the same parametrization as in the matter sector, shown in sec. 5.3.4.

5.3.3.2 Implementation in SARAH

It is possible to define the mixing in the gauge sector in the following way:

```
DEFINITION["Name of States"][GaugeSector]=
{{Vector Boson, {Generation 1, {{New 1a, Coeff 1a}, {New 2a, Coeff 2a},..},
                {Generation 2, {{New 1b, Coeff 1b}, {New 2b, Coeff 2b},..},
               ... },
 ...
 {Gaugino,      {Generation 1, {{New 1a, Coeff 1a}, {New 2a, Coeff 2a},..},
                {Generation 2, {{New 1b, Coeff 1b}, {New 2b, Coeff 2b},..},
               ... }};
```

Here, `New XY` is the name of the new mass eigenstates and `Coeff 1a` is the mixing element of the new and the old eigenstate. SARAH interprets this definition as a sum of sums:

$$V_i \to \sum_{n=1}^{N^2-1} \left(\delta n, i \sum_{m=1} c_{nm} \tilde{V}_{nm} \right) \tag{5.5}$$

V is the former eigenstate with generation index i running from 1 to N^2-1, c_{nm} are the coefficients for the new eigenstates $\tilde{V}_{nm}$.

Example We consider the electroweak symmetry breaking of the MSSM

$$W_1 = \frac{1}{\sqrt{2}} \left(W^- + W^{-*} \right) , \tag{5.6}$$

$$W_2 = i\frac{1}{\sqrt{2}} \left(W^{-*} - W^- \right) , \tag{5.7}$$

$$W_3 = \sin\Theta_W \gamma + \cos\Theta_W Z , \tag{5.8}$$

$$B = \cos\Theta_W \gamma - \sin\Theta_W Z , \tag{5.9}$$

This is done by

```
{VWB, {1,{VWm,1/Sqrt[2]},{conj[VWm],1/Sqrt[2]}},
      {2,{VWm,-\[ImaginaryI]/Sqrt[2]},{conj[VWm],\[ImaginaryI]/Sqrt[2]}},
      {3,{VP, Sin[ThetaW]},{VZ, Cos[ThetaW]}}},
{B,   {1,{VP, Cos[ThetaW]},{VZ, -Sin[ThetaW]}}}
```

in SARAH.

> In the definition of EWSB W^+ (`VWp`) can not be used! This would introduce a new complex field not related to W^- (`VWm`) and therefore change the degrees of freedom of the theory. Of course, the winos $\tilde{W}^+$ and $\tilde{W}^-$ are not related by complex conjugation, so two new fields `fWm` and `fWp` are used after mixing.

SARAH always handles the vector bosons for unbroken gauge theories as real particles. If the definition of the new mass eigenstates is free of complex parameters (like in the case of photon and Z) SARAH handles also this new eigenstates as real parameters.

5.3.4 Particle Mixings

Symmetry breaking or also bilinear terms in the superpotential lead to a rotation of the former (gauge) eigenstates to new mass eigenstates. The definition of these rotations depends on the fact, if the corresponding mass matrix is hermitian or not.

5.3.4.1 Introduction

Properties of hermitian mass matrices In the hermitian case, the Lagrangian has the form

$$\mathcal{L}_{Mass} = \phi^\dagger M \phi. \tag{5.10}$$

The matrix M can be diagonalized by a matrix U:

$$M_{Dia} = U^{-1} M U. \tag{5.11}$$

The eigenvalues of the mass matrix M are the masses of the new mass eigenstates ψ, which are related to the former eigenstates by

$$\psi^i = U^{ij} \phi_j \tag{5.12}$$

Properties of non-hermitian mass matrices If the mass term in the Lagrangian is built by two vectors ϕ_1, ϕ_2, i.e.

$$\mathcal{L} = \phi_1^T M \phi_2 \ , \tag{5.13}$$

two mixing matrices are needed to diagonalize the mass matrix:

$$M_{Dia} = V^{-1} M U. \tag{5.14}$$

The two mixing matrices V and U diagonalize the matrices MM^T and $M^T M$

$$M^2_{Dia} = V^{-1} M M^T V, \qquad M^2_{Dia} = U^{-1} M^T M U \tag{5.15}$$

and connect the new eigenstates $\vec{\psi}_1$ and $\vec{\psi}_2$ with the old ones by

$$\psi_1^i = V_{ij} \phi_1^j, \qquad \psi_2^i = U_{ij} \phi_2^j \tag{5.16}$$

5.3.4.2 Implementation in SARAH

Both type of mixings are defined due to

```
DEFINITION["Name of States"][MatterSector] = { ... }
```

SARAH differs between hermitian and non-hermitian mixings by the form used for the definition. Note, that first all generations of one particle in the basis are inserted, before the next particle follows.

5.3.4.2.1 Hermitian mixings in SARAH

The form for defining hermitian rotations is

```
{{List of Old Eigenstates},{Name of New Eigenstates, Name of Mixing Matrix}}
```

a) First, a list of the names of old eigenstates is given.

b) The name of the new eigenstates must be given.

c) The name of the mixing matrix must be given.

Examples The mixing in the down-squark sector is given by

```
{{SdL, SdR}, {Sd, ZD}}
```

This means, the three left d-squarks `SdL` and the three right d-squarks `SdR` mix to new eigenstates called `Sd` with generation index running from 1 to 6. The mixing matrix is called `MD`. The above statement leads to the following relation between the states

$$\tilde{d}^i = \sum_{j=1}^{3} Z^D_{i,j} \tilde{d}^j_L \tag{5.17}$$

$$\tilde{d}^{i+3} = \sum_{j=1}^{3} Z^D_{i+3,j} \tilde{d}^j_R \tag{5.18}$$

In the flavor conserving case, this matrix is reducible. Therefore, `SARAH` checks all matrices for reducibility and sets the non-block elements automatically to zero.

5.3.4.2.2 Non-hermitian mixings in SARAH

Non-hermitian rotations are defined in `SARAH` by

```
{{{First Basis},{Second Basis}},{{First States,First Matrix},{Second States,Second Matrix}}}
```

Let us clarify this convention by an example.

Example We consider the chargino sector in the MSSM. This mixing is specified by

```
{{{fWm, FHdm}, {fWp, FHup}}, {{Lm,Um}, {Lp,Up}}}
```

This means that the gauge eigenstates $\tilde{W}^-$ (`fWm`) and $\tilde{H}_d^-$ (`FHdm`) mix to the negative charged mass eigenstates λ^- (`Lm`), while $\tilde{W}^+$ (`fWp`) and $\tilde{H}_u^+$ (`FHup`) form the new eigenstates λ^+ (`Lp`). The new and old eigenstates are connected by the mixing matrices U^- (`Um`) and U^+ (`Up`).

$$\begin{pmatrix} \lambda_1^- \\ \lambda_2^- \end{pmatrix} = U^- \begin{pmatrix} \tilde{W}^- \\ \tilde{H}_d^- \end{pmatrix}, \qquad \begin{pmatrix} \lambda_1^+ \\ \lambda_2^+ \end{pmatrix} = U^+ \begin{pmatrix} \tilde{W}^+ \\ \tilde{H}_u^+ \end{pmatrix} \tag{5.19}$$

5.3.5 Gauge fixing terms and ghost interactions

5.3.5.1 Introduction

As explained in app. B, the general form of a gauge fixing term in R_ξ-gauge is

$$\mathcal{L}_{GF} = -\frac{1}{2R_\xi} \sum_a |f^a|^2 \tag{5.20}$$

with some gauge fixing functions f^a.

5.3.5.2 Implementation in SARAH

`SARAH` calculates the ghost interactions from the gauge fixing terms. It is possible to define the gauge fixing part of the Lagrangian for the different eigenstates by

```
DEFINITION["Name of States"][GaugeFixing]=
{{Function, Prefactor}, ... };
```

`Function` is the f of eq. (5.20), and the corresponding factor is `Prefactor`. If the gauge fixing functions involve derivatives of gauge bosons,

```
Der["Gauge Boson"]
```

has to be used. For Goldstone bosons, the internal name in `SARAH` must be used.

Examples

a) The gauge fixing term for the color group in R_ξ gauge is:

$$\mathcal{L}_{GF} = -\frac{1}{2\xi_g}|\partial_\mu g|^2 \tag{5.21}$$

This is given in the model file by

```
DEFINITION[GaugeES][GaugeFixing]=
{{Der[VG], -1/(2 RXi[G])},...};
```

b) The gauge fixing term corresponding to the Z-Boson after EWSB is (see app. B.19).

$$\mathcal{L}_{GF} = -\frac{1}{2\xi_Z}\left(\partial^\mu Z_\mu + \xi_Z M_Z G^0\right)^2 \tag{5.22}$$

The corresponding Goldstone boson is in `SARAH` the first generation of the CP-Odd Higgs (see app. F), i.e. `Ah[{1}]`. Therefore, we can write the gauge fixing term as

```
DEFINITION[EWSB][GaugeFixing]=
{{Der[VZ] + Mass[VZ] RXi[Z] Ah[{1}],  - 1/(2 RXi[Z])},...};
```

5.3.6 Additional couplings or redefinition of existing couplings

5.3.6.1 Introduction

Sometimes, it might be necessary to define interactions in the Lagrangian, which can not be derived from the superpotential or the kinetic interaction. Furthermore, it might be necessary to change the properties of some vertices by hand. For example, integrating out particles might most likely spoil SUSY. Therefore, supersymmetric relations are not longer valid: the standard model Higgs self couplings are free parameters, while they are fixed in SUSY by the gauge and Yukawa couplings.

5.3.6.2 Implementation in `SARAH`

Both, defining new interactions and changing existing ones, is done in `SARAH` with one statement. For each set of eigenstates, new or changed terms can be defined separately by declaring them in the model file via

```
DEFINITION["Name Of Eigenstates"][Additonal] = {
{Lag, {Options}},
  ... };
```

`Lag` is a Lagrangian which is added to the complete Lagrangian, so it must has mass dimension 4. A rather short notation can be used, only some points have to be considered

- Fields are separated by dots
- Weyl fermions are used
- All indices are automatically added and contracted

The new couplings are handled in a similar way as the couplings of the superpotential: tensor indices are added automatically and they are assumed to be complex. Further assumptions about the coupling can be made in `parameters.m`.
The manually defined terms will be handled like every other term of the Lagrangian, i.e. they are affected by rotations and replacements if the eigenstates are changed.
The two possible options are:

a) `AddHC -> True/False`: Defines if the hermitian conjugated of this term is not added to the Lagrangian

b) `Overwrite -> True/False`: Defines if exisiting couplings involving the same fields are overwritten.

Example

a) Define new terms A mixed soft-breaking term of the form

$$m^2_{\tilde{l}H_d}(\tilde{l}^* H_d + \tilde{l} H_d^*) \tag{5.23}$$

is added to the Lagrangian of the gauge eigenstates by

```
DEFINITION[GaugeES][Additional] =
{{mlHd2 conj[Sl].SHd, {Overwrite->False, AddHC->True}}};
```

b) Adding additional terms to existing couplings With

```
DEFINITION[EWSB][Additional] =
{{1/24 Kappa hh.hh.hh.hh, {Overwrite->False, AddHC->False}}};
```

the Higgs self couplings receive a additional contribution:

$$\Gamma_{h^4}\ (a_i g_i^2 + b_i Y_i^2) \quad \rightarrow \quad \Gamma_{h^4}\ (a_i g_i^2 + b_i Y_i^2 + \kappa) \tag{5.24}$$

c) Overwriting existing terms To overwrite the former expressions for the Higgs self interactions,

```
DEFINITION[EWSB][Additional] =
{{LagNew, {Overwrite->True, AddHC->False}}};
LagNew = 1/24 Kappa hh.hh.hh.hh + 1/24 Lambda hh.hh.Ah.Ah;
```

is used. This has the following effect:

$$\Gamma_{h^4}\ (a_i g_i^2 + b_i Y_i^2) \quad \rightarrow \quad \Gamma_{h^4}\ \kappa \tag{5.25}$$

d) Interactions involving derivatives For interactions involving derivatives, `Der` is used

```
DEFINITION[EWSB][Additional] =
{{Kappa Der[SHd,lor3].conj[SHd].VB,{Overwrite->False, AddHC->True}}};
```

e) Matter interactions of the standard model This method can be used to define the matter interactions of the standard model

```
SuperPotential = {};
DEFINITION[GaugeES][Additional]= {
{LagHC, {Overwrite->True, AddHC->True}},
{LagNoHC,{Overwrite->True, AddHC->False}}
};
LagNoHC = Mu conj[SH].SH + 1/24 Lambda1 conj[SH].SH.conj[SH].SH;
LagHC = - Yd conj[SH].Fq.conj[FdR] - Ye conj[SH].Fl.conj[FeR] + Yu SH.Fq.conj[FuR];
```

5.4 Effective or non-supersymmetric theories

It is easy in SARAH to integrate particles out of the spectrum to get an effective theory, or just to delete this particle to get a non-supersymmetric limit of the model.

5.4.1 Integrating out particles

5.4.1.1 General

If in a theory very heavy particles Φ exist, they are no physical degree of freedom if the energy scale is below the mass of the particle. In this case, an effective theory for the lighter fields ϕ is derived by integrating out the heavy states

$$L(\phi)_{eff} = \int L(\phi, \Phi) d\Phi \tag{5.26}$$

This procedure will lead to higher dimensional operators like the four fermion interaction in Fermi's theory.

5.4.1.2 Implementation in SARAH

Integrating out particles in SARAH is easy: the heavy particles, which should be integrated by, are added to IntegrateOut:

```
IntegrateOut = {Particle 1, Particle 2,...}
```

Here, `Particle 1` can be the component name of a particle, e.g `SdR` for all right d-squarks. Moreover, if the superfield name is used, all corresponding component fields are integrated out. If only specific generations of one field should be integrated out, the assignment is

```
IntegrateOut = {Particle[{first,last}],...}
```

Here, `first` is the first generation which should be integrated out and `last` is the last one.

> The list `IntegrateOut` can consist of particles belonging to different eigenstates. All particles are integrated out at the first time they do appear.

Example

a) To get an effective theory by integrating out the gluino, use

```
IntegrateOut[[1]] = fG;
```

b) RGE running of soft-breaking masses leads to the effect that the first two generations of sleptons are heavier than the third one. Therefore, building an effective theory with only the third generation of squarks is done by

```
IntegrateOut = {SdR[{1,2}],SuR[{1,2}],SdL[{1,2}],SuL[{1,2}]};
```

5.4.2 Deleting particles

Deleting particles can be done in the same way as integrating them out. The difference is that there are no effective operators are calculated. Deleting is therefore faster and should be used, if the higher dimensional operators are not needed.
Deleting particles is done by

```
DeleteParticle={List Of Particles};
```

The usage of `DeleteParticles` is the same as of `IntegrateOut`.

5.5 Definition of Dirac spinors

Event generators and programs for calculating Feynman Diagrams are normally written for Dirac spinors, but `SARAH` does all internal calculations for Weyl spinors. Therefore, it is necessary to define, how the Weyl spinors combine to Dirac spinors by

```
dirac[[i]] = {Dirac Spinor, Weyl 1, Weyl 2}
```

`Dirac Spinor` is the new name of the Dirac spinor, while `Weyl 1` and `Weyl 2` the names of the Weyl spinors building the left respectively right component of the Dirac spinor.

Example The electron `Fe` is built from the components `FeL` and `FeR` by

```
dirac[[i]] = {Fe, FeL, FeR}
```

while the neutralinos `Chi`, which are Majorana particles, consists only of the mass eigenstates `L0`

```
dirac[[i]] = {Chi, L0, conj[L0]}
```

Chapter 6

Details about the different output

How to generate output files by using the command `ModelOutput` was already explained in section sec. 3.7. Here, we want to give some further information about the different routines.

6.1 LaTeX

6.1.1 Writing a LaTeX file

It is possible to write a LaTeX file with all information about the mode by using

```
ModelOutput[Eigenstates, WriteTeX->True];
```

This calculates first all interactions for the eigenstates. If this was already done before, it is also possible to use

```
MakeTeX[Options];
```

There are different Tex-files produced containing the following information:

a) List of the fields

b) Important parts of the Lagrangian

c) Mass Matrices, if there is any mixing

d) Renormalization Group Equations

e) Tadpole equations, if there is a broken gauge symmetry

f) One-loop self energies and tadpole equations

g) All interactions

6.1.2 Options

The options are

a) `FeynmanDiagrams`, Values: `True` or `False`, Default: `True`
Defines, if the Feynman diagrams for all interactions should be drawn.

b) `EffectiveOperators`, Values: `True` or `False`, Default: `True`
Defines, if the higher dimensional operators should be included in the LaTeXfile. By default, this are only the vertices involving up to four particles. For switching on six particle interactions,`SixParticleInteractions` is used.

c) `SixParticleInteractions`, Values: `True` or `False`, Default: `False`
Defines, if also the six-particle interactions should be added to the LaTeX output

d) `ShortForm`, Values: `True` or `False`, Default: `False`
Defines, if a shorter notation for the vertices should be used

e) `WriteSARAH`, Values: `True` or `False`, Default: `False`
Defines, if the names and parameters used in `SARAH` should be written

6.1.3 Making the pdf File

The Tex files are saved in the directory

```
../Output/"Name of Model"/"Name of Eigenstates"/TeX
```

and the main file is `"Name of Model" <> "Name of Eigenstates" <>.Tex`. All other files are included in this file by using the `input`-command of LaTeX. If `Diagrams` is set to `True`, the following steps must be done for generating an pdf document including the diagrams:

a) First, compile the Tex file, e.g. `pdflatex model.tex`

b) Go to the directory `Diagrams` and compile every `.mp` file with `mpost`. This is done under Linux and under Windows with

```
mpost FeynmanDiaX.mp
```

It is also possible to apply the `mpost` command on all `.mp`-files at once by using

```
find . -name "*.mp" -exec mpost {} \;
```

c) After generating all diagrams, go back and compile the `.tex`-file again by using `pdflatex`.

To simplify this procedure, `SARAH` will write a shell script in the Tex-output directory which does exactly these three steps. It can be started under Linux with

```
./MakePDF.sh
```

or under Windows with

```
MakePDF.bat
```

It is possible that the script must be first declared is executable in Linux by

```
chmod 755 MakePDF.sh
```

6.2 FeynArts

6.2.1 Generate model files for FeynArts

A model file for \FeynArts is created by

```
ModelOutput[Eigenstates, Write\FeynArts->True]
```

or, if ModelOutput was used before and the vertices are already calculated, by

```
Make\FeynArts
```

The following things are done:

a) A list of all particles in the models is generated. The particles in FeynArts are named

(a) S[X]: For scalars, with some integer X

(b) F[X]: For fermions, with some integer X

(c) V[X]: For vector bosons, with some integer X

(d) U[X]: For ghosts, with some integer X

X can be defined in the particle definitions file of SARAH or is chosen automatically. \FeynArts also supports labels for particles which are easier to read for humans eyes by using a TeX-like output. The label for each particle is generated from the defined LaTeXname by SARAH.

b) A list with all appearing indices is written. These are the indices for generation and non-abelian charges. In contrast to the originally model file for MSSM of FeynArts, every particle gets an independent generation index.

c) The list with interactions is written. If the theory contains several non-abelian, unbroken gauge groups, the generators of these gauge groups will appear in the vertices. By default, the generators of $SU(3)$ are associated to SUNT and automatically simplified by using \FormCalc [22]. The algebra for the generators of another dimension must be done separately.

6.2.2 Dependences, numerical values and special abbreviations for FormCalc

A second file is generated by SARAH together with the model file for FeynArts: Substitutions-<> "Name of Eigenstates" <>.m . This file contains additional information which might be useful for calculating diagrams:

a) Replacement rules with the defined dependences in SARAH, see sec. 4.2.3:

(a) Dependences

(b) DependencesOptional

(c) DependencesNum

b) The definitions of the masses: Masses\SARAH

c) The numerical values for the parameters in SARAH: NumericalValues

d) Special abbreviations for FormCalc like those are also defined for the MSSM and SM in the FormCalc package:

(a) A complex conjugation is replaced by C: `Conjugate[X] -> XC`

(b) A square is replaced by 2: `X^2 -> X2`

(c) The names of soft-breaking couplings are merged: `T[X] -> TX`

It is suggested to use this definitions to speed up the calculations with `FormCalc`.

6.2.3 Higher Dimensional Operators

`\FeynArts` does not support higher dimensional operators because the corresponding Lorentz structures are not given in `Lorentz.gen`. Therefore, such operators are not included in the model file even if an effective theory is considered.

6.3 CalcHep/CompHep

6.3.1 Generate model files for CalcHep and CompHep

To generate model files for `CalcHep` and `CompHep`,

```
ModelOutput["Name of Eigenstates", WriteCHep->True]
```

is used.If the vertices have been calculated already by `MakeVertexList`,

```
MakeCHep[options]
```

can be used. The possible options are

a) `FeynmanGauge`, Values: `True` or `False`, Default: `True`
By setting to `True`, the interactions of the Goldstone bosons are written in the interaction file.

b) `CPViolation`, Values `True` or `False`, Default: `False`
By setting to `True`, the possibility of CP violation is included in the model files, see sec. 6.3.5.

c) `ModelNr`, Values Integer, Default: `1`
The number added to file names, see next section.

d) `CompHep`, Values: `True` or `False`, Default: `False`
By setting to `True` the model files are written in the `CompHep` format.

e) `NoSplittingWith`, Values: Particles List, Default: `{}`
If one of the given particles appears in a four-point interaction, the interaction is not split using auxiliary fields

f) `NoSplittingOnly`, Values: Particles List, Default: `{}`
If all particles of a four-point interaction appear in the given list, the interaction is not split using auxiliary fields

g) `UseRunningCoupling`, Values: `True` or `False`, Default: `False`
Defines, if the standard running of the strong coupling should be included in the model file.

h) `SLHAinput`, Values: `True` or `False`, Default: `False`
Defines, if parameter values should be read from a LesHouches input file, see sec. 6.3.6.

6.3.2 Model Files

The CalcHep/CompHep output of SARAH generates the following four files

a) prtclX.mdl: Contains all particles

b) lgrngX.mdl: Contains the interactions

c) varsX.mdl: Contains the numerical values of the variables

d) funcX.mdl: Contains dependences between the parameters

X is a number, which can be chosen by the option ModelNr.

6.3.2.1 Particles

First, there are stringent constraints on the naming of particles in CalcHep: only names up to four letters are allowed and also indices aren't supported. Therefore, it is not possible to use the SARAH internal definitions of particles. Thus, the names used for the model files are based on the defined OutputName of each particle in the following way

a) The basis of each name is the entry in OutputName in the particle file, see sec. 4.3

b) If the considered particle is not self-conjugated, for the anti particle the first letter is changed from upper to lower case or vice versa.

c) If there are more generations for one particle, the number of the generation is appended at the end of the name

d) If the defined R-parity is -1, a ~ is added to the beginning of the name to assign SUSY particles. In this way, it is possible to use the model files in MicrOmegas without the need of an additional list of all SUSY particles

The steps above are the standard procedure for all vector bosons, fermions and most scalars. Ghosts, Goldstone bosons and auxiliary fields handled in a different way. There are three different kind of ghosts. These are not written in the particle file, but appear in the Lagrangian file:

a) Faddeev-Popov Ghost: these are the well known Ghost derived from the gauge transformations of the gauge fixing term. The name in the model file is

```
"Name of Vector Boson".C
```

b) Goldstone Ghosts: these are just the Goldstone bosons 'eaten up' by the gauge particles. Their name is

```
"Name of Vector Boson".f
```

c) Tensor Ghosts: Is needed to express the four gluon interaction. The name is

```
"Name of Gluon".t
```

SARAH derives the name of Goldstone and Faddeev-Popov ghost automatically from the underlying vector boson, but the tensor ghost and its one interaction with two gluons is hard-coded.

The last kind of fields known by CalcHep/CompHep are auxiliary fields. Their purpose is explained in the next section, but their names are as follows

```
~OX
```

Here, `X` is a integer. The antiparticle, if it is not the same, is counted as X+1.

6.3.3 Auxiliary fields in CalcHep/CompHep

We mentioned in the last section that `CalcHep` and `CompHep` needs special auxiliary fields. The reason is that the color structure is implicit. Hence, interactions of four colored particles or two colored and two gluons suffer from an ambiguity. Therefore, these interactions are split in two three particle interactions by inserting auxiliary fields.
`SARAH` does a similar splitting for all interactions between four scalars by inserting auxiliary fields when calculating the F- and D-Terms. Also the interactions between two squarks and two gluons are split in two three particle interactions. The splitting can be suppressed for specific vertices by using `NoSplittingWith` or `NoSplittingOnly`.

6.3.4 Vertex functions

All interactions are parametrized by a variable in the Lagrange file. The values of these variables are defined by using the results of `SARAH` for the corresponding vertices. The following renaming had to be done:

a) Tensor indices are just added to the name, therefore all sums in the vertices had do be evaluated:

```
sum[i1,1,3,MD[1,i1]]      ->       MD11 + MD12 + MD13
```

Some values, which are known to be zero like in the flavor conserving case, are set to zero at this point.

b) Variables names, which are longer then six letters, are truncated.

c) All parameters are assumed to be real, i.e. complex conjugation is removed (see sec. 6.3.5).

d) The generators and structure constants of the strong interaction are removed because they are defined implicitly.

6.3.5 CP Violation

`CalcHep`/`CompHep` can't handle complex values in the function or vars file, but only in the Lagrange file. Therefore, all variables are by default assumed to be real, when `SARAH` writes the model file with default options. This can be changed by setting `CPViolation` to `True`. In that case, `SARAH` splits all parameters, which are not explicitly defined as real, in real and imaginary part:

```
X -> RX + i*IX
```

The real and imaginary part for every interaction is calculated using that splitting, and both parts are written separately in the Lagrange file:

```
v0001 -> Rv0001 + i*Iv0001
```

6.3.6 SLHA input

CalcHep supports the possibility to read LesHouches input files [23]. SARAH can write the corresponding definitions in the functions file of CalcHep. In this context, it is assumed that a LesHouches file called Spectrum.spc is located in the same directory. However, this can easily be adjusted manually.

6.3.7 What can be a problem...

We have made the following experiences by testing model files with CalcHep/CompHep:

a) A PDG number of 0 is not allowed for other particles than auxiliary particles.

b) In the vars file is no discrimination between small and capital letters. This must be taken into account by naming the mixing matrices and couplings in SARAH.

c) Higher dimensional operators are not supported

d) The color structure is implicit and indices are not supported in CalcHep/CompHep. Therefore, it is difficult to implement models with other unbroken non-abelian gauge groups than the color group.

e) Writing the output for models with particles appearing in a large number of generations and non reducible mixing matrices like in the flavor conserving MSSM, last very long, because all possible combinations of indices have to be written separately.

6.4 WHIZARD (in collaboration with Christian Speckner)

To generate model files for WHIZARD

```
ModelOutput["Name of Eigenstates", WriteWHIZARD->True]
```

is used.If the vertices have been calculated already by MakeVertexList,

```
MakeWHIZARD[options]
```

can be used. The possible options are

a) WriteOmega, Values: True or False, Default: True
Defines, if the model files for Omega should be written

b) WriteWHIZARD, Values: True or False, Default: True
Defines, if the model files for WHIZARD should be written

c) Exclude, Values: list of generic type, Default: {SSSS}
Defines, which generic diagrams are exlcuded when writing the model file

d) WOModelName, Values: string, Default: defined model name
Gives the possibility to change the model name

e) MaximalCouplingsPerFile, Values: Number, Default: 500
Defines the maximal number of couplings written in one file

f) Version, Values: Number, Default: 2.0.3
Defines the version of WHIZARD for which the model file is generated

g) ReadLists, Values: True or False, Default: False
Defines, if the information from a former evaluation should be used

6.5 Automatic output

It is possible, to generate the desired output automatically. To do that the array `MakeOutput` in the model file is used

```
MakeOutput = {{ Eigenstates 1 , List 1}, { Eigenstates 2, List 2}, ... };
```

`Eigenstates` fixes the eigenstates for which the output should be generated. The entries in `List X` can be the following:

a) TeX

b) `FeynArts`

c) CHep

Example To get a LaTeX- and `FeynArts` file for the eigenstates after electroweak symmetry breaking,

```
MakeOutput = {{EWSB, {TeX, \FeynArts}}};
```

is used.

Chapter 7

SPheno output

7.1 Introduction

SPheno [18] is a spectrum generator for the MSSM and several extensions of it. However, all models have in common that they are hard coded. SARAH should dramatically simplify the implementation of new models in SPheno by generating source code for all model dependent parts. This code can later on used with SPheno to calculate the mass spectrum and branching ratios of the new model.

The generation of the source code for SPheno is started via

```
MakeSPheno[Options]
```

The different options are:

- Eigenstates->Name of Eigenstates.. If not given, by default the last entry of NameOfStates is used.
- ReadLists->True can be used if all vertices and RGEs have already been calculated for the model and the former results should be used to save time.
- InputFile. The name of the SPheno input file. If not defined, SPheno.m is used.

The generated source code is located in

```
/Directory of SARAH/Output/Model/Eigenstates/SPheno/
```

Generate a new directory in your SPheno directory with the same name as the name of the model and copy all files into that directory and use make to compile the new SPheno version. A new binary

```
SPheno <> ModelName
```

is generated in bin/.

7.2 Input file

For the SPheno output, a new file SPheno.m is needed to define the properties of the generated SPheno version. The file SPheno.m must be located in the same directory as the other models files for SARAH.

a) RenameParameters: Sometimes, it might be necessary to rename some parameters especially for the SPheno output, e.g. if the name is already used in SPheno like lambda. This is done in the following way:

```
RenameParamters = {{Lambda,LambdaNMSSM},{Kappa,KappaNMSSM},... }
```

First, the internal name in SARAH is given, afterwards the name is defined which should be used for the SPheno output.

b) MINPAR: A list of parameters which should be read from a LesHouches file by SPheno. For example:

```
MINPAR={m0,m12,A0,signMue,tanbeta};
```

The order of the parameters in Block MINPAR of the LesHouches file must be the same as in this array.

c) EXTPAR: It is also possible to define additional parameters for the block EXTPAR of the LesHouches input file by

```
EXTPAR = {{Nr1,Var1}, {Nr2,Var2},...};
```

First, the number in the block is defined, afterwards the variable. For instance, to give three additonal VEVs as input, use

```
EXTPAR = {{100, v1},
          {101, v2},
          {102, v3}};
```

d) ParametersToSolveTadpoles: The list of parameters, whose numerical values should be fixed the tadpole equations. For example, to use the standard choice in the MSSM μ, B_μ, the entry reads:

```
ParametersToSolveTadpoles = {\[mu], B[\mu]};
```

SARAH uses the Solve command of Mathematica to solve the tadpole equations for the given set of parameters. If the solution is not unique because a parameter X appears squared, SARAH solves the equations for the absolute squared. The phase is defined by the automatically generated variable SignumX, which is expected to be given as input.

e) UseGivenTapdoleSolution: In cases, in which Mathematica won't find an analytical solution for the tadpole equations for the given set of parameters, this variable has to be set to True and an approximated solution can be given. These solutions are defined by

- SubSolutionsTadpolesTree: For the solution at tree level

  ```
  SubSolutionsTadpolesTree = {x1 -> sol1, x2 -> sol2,...}
  ```

- SubSolutionsTadpolesLoop: For the solutions of the one loop corrected tadpoles. The one loop corrections to the different VEVs have to be named Tad1Loop[i].

f) RenormalizationScaleFirstGuess: For a first running of the RGEs, before any mass has been calculated by SPheno, the renormalization scale can be defined by this entry. For example, for a mSugra scenerio

```
RenormalizationScaleFirstGuess = m0^2 + 4 m12^2;
```

This affects the running only if SPA conventions are disabled and now fixed value is given in the LesHouches file.

g) `RenormalizationScale`: For all further runs, another renormalization scale can be given, e.g.

```
RenormalizationScale = MSu[1]*MSu[6];
```

h) Two loop contributions to the Higgs masses: if the Higgs sector of the model is the same as for the MSSM, the original SPheno routines for calculating the two loop tadpole equations and two loop self energies to the the scalar and pseudo scalar Higgs can be implemented by setting

```
UseHiggs2LoopMSSM = True;
```

i) Boundary Condition: It is possible to define boundary conditions at three different scales:

- Electroweak scale: `BoundaryEWSBScale`
- SUSY scale: `BoundarySUSYScale`
- GUT scale: `BoundaryHighScale`

In addition, if thresholds are involved, boundary conditions can be set at the threshold scale. See section 7.3.
All boundaries are defined by a two dimensional array. The first entry is the name of the parameter, the second entry is the used condition at the considered scale. The condition can be ...

- ... an input parameter, e.g.

  ```
  {MassB, m12}
  ```

- ... the product of parameters, e.g.

  ```
  {TYd, Azero*Yd}
  ```

- ... a diagonal matrix, e.g.

  ```
  {md2, DIAGONAL m0^2}
  ```

- ... the solution of the tadpole equation, e.g.

  ```
  {mHd2, TADPOLE}
  ```

- ... matrix multiplications or the inverse of a matrix, e.g.

  ```
  {X, MatMul2[A,InverseMatrix[B], FortranFalse]};
  ```

For the matrix multiplication `MatMul2` has to be used. The third argument assigns, if only diagonal elements (`FortranTrue`) should be considered or not (`FortranFalse`). **All boundary condtions are overwritten, if in the LesHouches input a value for the parameter is given**. For example, with

```
Block mHd2IN   #
   10000.000    #
Block mHu2IN   #
   20000.00    #
```

The Higgs soft breaking masses at the GUT scale are not set to m_0^2, but to the given values.

Several sets of boundary conditions To implement different version of one model which differ only by the used boundary conditions, `BoundaryEWSBScale`, `BoundarySUSYScale`, `BoundaryHighScale` can be a nested list. For instance,

```
BoundarySUSYScale[[1]] = {
  {KappaNMSSM, KappaInput},
  {LambdaNMSSM, LambdaInput}};
BoundaryGUTScale[[1]] = {};

BoundarySUSYScale[[2]] = {};
BoundaryGUTScale[[2]] = {
  {KappaNMSSM, KappaInput},
  {LambdaNMSSM, LambdaInput}};
```

In the first case, the input values for λ and κ are taken at the SUSY scale, in the second on at the GUT scale. To let `SPheno` know which set of boundary conditions should be used for a run, the flag 2 in `MODSEL` is used:

```
Block MODSEL #
2  X  # This used the X. set of boundary conditionx.
```

The default value is 1.

j) Lists for calculating decay widths:

- `ListDecayParticles`: List of particles for whicht the two-body decays should be calculated. This can be a list of particles using the names inside `SARAH`, e.g.

  ```
  ListDecayParticles = {Sd,Su,Se,hh,Ah,Hpm,Chi};
  ```

  or just `Automatic`. If `Automatic` is used, the widths of all particles not defined as standard model particles as well as for the top quark are calculated.

- `ListDecayParticles3B`; Three body decays of fermions. This can be a list with the names of the particles and the corresponding files names, e.g.

  ```
  ListDecayParticles3B =  {{Chi,"Neutralino.f90"},
                           {Cha,"Chargino.f90"},
                           {Glu,"Gluino.f90"}};
  ```

  or just `Automatic`. If `Automatic` is used, the widths of all fermions not defined as standard model particles are calculated. The automatic generated file names are `ParticleName.f90`.

k) Ordering of masses: normally, all masses are ordered in `SPheno` by their mass. However, it might be necessary in some cases to give a condition for ordering of some masses. This is for example the case, if several massless CP odd particles at tree level exist to assign the goldstone boson. For this purpose, a condition can be defined by using

```
ConditionForMassOrdering = { {Particle, Condition}, ... };
```

The condition has to be Fortran source code and is added to the corresponding routine. For instance, a conidtion for the NMSSM would read

```
ConditionForMassOrdering={
{Ah,
"If ((Abs(ZA(1,3)).gt.Abs(ZA(2,3))).And.(MAh2(1).lt.1._dp) &
        &.And.(MAh2(2).lt.1._dp)) Then \n
   MAh2temp = MAh2 \n
   ZAtemp = ZA \n
   ZA(1,:) = ZAtemp(2,:) \n
   ZA(2,:) = ZAtemp(1,:) \n
   MAh2(1) = MAh2temp(2) \n
   MAh2(2) = MAh2temp(1) \n
End If \n \n"}
};
```

This condition checks, if two light pseudo scalars in the spectrum. If this is the case, it uses as goldstone boson the not singlet-like particle.

Information about particles and parameters

SARAH needs some information about the physical meaning of some particles and parameters. These information are used to calculate the gauge and Yukawa couplings at the electroweak scale, calculate the CKM matrix, use the correct on-shell masses, etc. All definitions are done by the description statement in the parameters and particles file.

a) The following particles are needed:

- Leptons
- Down-Quarks
- Up-Quarks
- Photon
- Gluon
- W-Boson
- Z-Boson
- Up-Squarks
- Higgs
- Pseudo-Scalar Higgs
- Charged Higgs
- Neutrinos
- Sneutrinos

b) The following parameters have to be defined:

- Up-Yukawa-Coupling
- Down-Yukawa-Coupling
- Lepton-Yukawa-Coupling
- Hypercharge-Coupling

- Left-Coupling
- Strong-Coupling
- Up-Squark-Mixing-Matrix
- Down-Squark-Mixing-Matrix
- Left-Lepton-Mixing-Matrix
- Right-Lepton-Mixing-Matrix
- Left-Down-Mixing-Matrix
- Right-Down-Mixing-Matrix
- Left-Up-Mixing-Matrix
- Right-Up-Mixing-Matrix
- Weinberg-Angle
- Down-VEV
- Up-VEV
- Scalar-Mixing-Matrix
- Pseudo-Scalar-Mixing-Matrix
- Softbreaking right Down-Squark Mass
- Softbreaking right Up-Squark Mass
- Softbreaking left Slepton Mass
- Softbreaking right Slepton Mass
- Softbreaking left Squark Mass
- Trilinear-Up-Coupling
- Trilinear-Down-Coupling
- Trilinear-Lepton-Coupling

Mixing angles in gauge sector

The mixings in the gauge sector are normally expressed by some angles, e.g, Θ_W in SM/MSSM. Since the solution of trigonometrical functions are not unique, SARAH doesn't calculate the analytical expressions for those angels. These formulas have been defined using the statement `Dependence` or `DependenceNum` in the parameter file, e.g.

```
{ThetaW,   { Description -> "Weinberg-Angle",
             DependenceNum -> ArcCos[g2/Sqrt[g1^2+g2^2]],
  }}
```

7.3 Including Thresholds

It is possible to include thresholds in the RGE running.

7.3.1 Same gauge structure for all scales

If all scales have the same gauge structure, it is possible for SARAH to derive the RGEs for all scales from the RGEs for the highest scale by performing the following steps.

- The generations of the heavy fields are variables, which are dynamically adjusted.
- The parameters which involve heavy fields are set to zero below the corresponding scale
- It is assumed, that the masses of the scalar and fermionic component of a heavy superfield are the same. These masses are given by a bilinear superpotential term.

To include such threshold effects, the following steps have to be performed:

a) The heavy fields must be deleted in the model file of SARAH:

```
DeleteFields = {...}
```

This ensures, that the decays, loop corrections, etc. at the SUSY scale calculated by SPheno receive no contributions from the heavy fields

b) The thresholds have to defined in SPheno.m :

```
Thresholds = {{Scale1, {HeavyFields1}},
              {Scale2, ... }};
```

For all scales an entry in the array Thresholds has to be added. Each entry defines first the threshold scale, at second position a list with the heavy superfields is given. Also specific generations for a superfield can be given.

It is possible to define boundary conditions at each threshold scale for running up and down separately:

```
BoundaryConditionsUp[[x]] = { ...}
BoundaryConditionsDown[[x]] = { ...}
```

Threshold corrections Using 2-loop RGEs demands 1-loop boundary condition. Therefore, at each threshold scale the one loop threshold effects for gauge couplings and gauginos calculated.

Examples This method is used to generate a version of SPheno for the seesaw type II and type III. The entries in"SPheno.m"are the following

a) Seesaw II:

```
 Thresholds={
{Abs[MTMIN],{s,sb,t,tb,z,zb}}
};
```

b) Seesaw III:

```
Thresholds={
{Abs[MWM3IN[1,1]],{Hx3[1],Hxb3[1],Hg3[1],Hb3[1],Hw3[1]}},
{Abs[MWM3IN[2,2]],{Hx3[2],Hxb3[2],Hg3[2],Hb3[2],Hw3[2]}},
{Abs[MWM3IN[3,3]],{Hx3[3],Hxb3[3],Hg3[3],Hb3[3],Hw3[3]}}
};
```

7.3.2 Different gauge structures

If the gauge structure at the different scales are different, each set of RGEs has to be calculated separately. Afterwards, this information has to be combined to one consistent version of SPheno. The final SPheno version includes routines for calculating finite shifts for the gauge couplings, gaugino and scalar mass parameters.

For doing this, the following steps are necessary:

a) For each regime a separate model file for SARAH has to be created

b) These model file have to be saved in the subdirectories Regime-1,Regime-2, ... beginning with the highest scale

c) The SPheno input file for the higher scales must provide the following information:

- IntermediateScale = True
- RegimeNr = X
- A list of the heavy fields, which should be integrated out:

  ```
  HeavyFields = {...};
  ```

d) All necessary information for combining the regimes to one SPheno is given in SPheno.m of the lowest scale.

- IntermediateScale = False
- RegimeNr = X
- The threshold scales: ThresholdScales = ...
- The boundary conditions for running up and down at each threshold scale:

  ```
  BoundaryConditionsUp[[x]] = { ...}
  BoundaryConditionsDown[[x]] = { ...}
  ```

  In the boundary conditions index1, index2, ... can be used for defining sums over indices.
- The usual information for SPheno, defined in the sec. 7.2.

When starting the SPheno output of the lowest scale, automatically all other scales are evolved. For the communication between the scales and for supervising the progress, a directory Meta in the model directory is created. In this directory, the screen output of Mathematica for each model is saved (Output-Regime-X.m). The necessary information for writing the combinded source code is saved in Regime-X.m.

Running of Vacuum Expectation Values (VEVs) If the same name of a VEV is appearing at two different scales, it is assumed, that it is really the same VEV and that the value of the VEV is given at the higher scale as input. The RGEs for this VEV are added automatically and used to get a consistent value for the VEV at the lower scale.

7.4 Generated Files for SPheno

All routines generated by SARAH are strongly inspired by the intrinsic routines of SPheno.

7.4.1 RGEs.f90

In principle, SARAH generates three sets of RGEs:

- Running of all parameters from the GUT scale to the SUSY scale
- Running of all parameters and VEVs from SUSY scale to electroweak scale (and back)
- Running of a minimal set of parameters from electroweak scale to GUT scale: The sets for running up and down can be defined in the two arrays LowScaleParameter and HighScaleParameter. More comfortable is the automatic derivation of the subsets, if these two arrays are not defined. The sets are derived in the following way:

  a) The starting is ...
  - ... IntermediateScale = False: The gauge couplings and Yukawa couplings are included
  - ... IntermediateScale = True: The content of MinimalParameterSetUp

  b) If the parameters from the tadpole equations influence the running of other parameters, also these parameters are added

  c) Afterwards, all parameters involved in the running of the so defined set are added

  d) This is repeated until there is no change in the set of parameters

For each set of RGEs the following function are generated:

- ParametersToGXX: Saves the parameters in a vector of length XX
- GToParametersXX: Extracts the parameters from a vector of length XX
- rgeXX: Definition of all β-functions.

Before writing the β-functions calculated by SARAH to the file, the following simplifications are done:

- All matrix multiplications are replaced by constants, which are calculated at the beginning of each routine
- All powers of numbers/parameters are replaced by a constants, which are calculated at the beginning of each routine

7.4.2 Couplings.f90

All vertices calculated by SARAH are written to this files. Also functions for calculating different subsets of these couplings are generated:

- AllCouplings: All couplings for the mass eigenstates
- CouplingsForSMfermion: Couplings involved in the one loop self energy of SM fermions
- CouplingsForVectorBosons: Couplings involved in the one loop self energy of SM vector bosons
- CouplingsForTadpoles: Couplings involved in the calculation of the one loop tadpoles
- CouplingsForLoop: Couplings needed for the one loop self energies of all particles: mixing matrices of external particles replaced by identity matrix.

7.4.3 SusyMasses.f90

This file contains functions to calculate all running masses at tree level. This is done in t'Hooft gauge: The masses of all Goldstone bosons are replaced by the masses of the corresponding vector boson. This works only, if the gauge fixing terms in the model file for SARAH have been added!

7.4.4 LoopMasses.f90

This file contains the routines for calculating the one loop contributions to the tadpoles and the self energies:

- `OneLoopMasses`: Main routine for calling all other functions
- `TadpolesOneLoop`: Calculates the one loop tadpoles.
- `Pi1LoopX`: One loop self energy for particle `X`: scalars or vector bosons (transverse part)
- `Sigma1LoopX`: Calculates $\Sigma_L, \Sigma_R, \Sigma_S$ for fermions
- `OneLoopX`: Calculates the one loop corrected mass for particle `X` in an iterative way

This is done again in t'Hooft gauge.

7.4.5 BranchingRatios.f90

Contains the routine for calculating the branching ratios for all particles. Checks are done, if three body decays for fermions are necessary or not. In addition, effective couplings of the Higgs bosons to two gluons or photons as well as to a real and virtual vector boson (W,Z) are calculated. For the couplings to real and virtual vector boson the SM values are used.

7.4.6 ModelData.f90

Contains the declaration of global variables and a function for initializing all variables with 0. Only masses of particles, which are integrated out, are initialized with 10^16.

7.4.7 LoopCouplings.f90

Routines for calculating the following couplings:

- Corrections to SM gauge couplings due to SUSY particles and top quark (`AlphaSDR, AlphaEwDR`)
- Effective couplings of scalar/pseudoscalar Higgs to SM fermions (e.g. `CouphhtoFe`)
- `DeltaVB`: Loop correction to δ_{VB} ($\mu \to e\nu_\mu\bar{\nu}_e$)

7.4.8 SugraRuns.f90

Main routines for calculating the (s)particle spectrum:

- `FirstGuess`: Calculates a first, approximate spectrum as starting point
- `SugraRuns`: Calculates in an iterative way the spectrum

- `BoundaryEW`: Calculates the starting point of the running for gauge and Yukawa couplings from electroweak data
- `BoundaryHS, BoundarySUSY`: Sets the boundary conditions at the GUT and SUSY scale.
- `BoundaryConditionsUp,BoundaryConditionsDown`: Applies the boundary conditions when running up/down at the threshold scales

7.4.9 `SusyDecays.f90`

Calculates all two body decays by using the phase space functions of `SPheno`. If scalar and pseudo scalar Higgs as well as the quarks are defined in the particles file by the corresponding `Description` statement, the one loop corrections to the decays $H, A \to q\bar{q}$ of $O(\alpha_s)$ are added.

7.4.10 Three Body Decays, e.g. `Glu.f90`

Contains all necessary routines to calculate the branching ratios and decay width of a fermion to three other fermions. All possible processes and diagrams are generated by `SARAH` and mapped to the phase space function of `SPheno`.

7.4.11 `InputOutput.f90`

This file contains the routines for reading the `LesHouches` input file and writing the output `SPheno.m`.

7.4.11.1 Input

The following changes to the standard LesHouches/`SPheno` input have been made:

- `Block MODSEL`: Since the generated `SPheno` works just for one model, the flag `1` is ignored
- `Block SPHENOINPUT`: New flags have been added:
  - `50`: If set to `0`, majorana fermions are not rotated to make all masses positive
  - `51`: If set to `0`, the parameters $Y_u, Y_d, T_u, T_d, m_q^2, m_d^2, m_u^2$ are not rotated in SCKM basis in `SPheno.spc`
  - `52`: Negative mass squares of particles are ignored and those masses are set back to zero.

7.4.11.2 Output

Routines for writing the following information to the LesHouches spectrum file are generated:

- Parameters at the SUSY scale:
  As block names for the output, the entries in `LesHouches` in the parameter file of `SARAH` are used. If this entry is missing, the name for the block is generated automatically.
- Masses at the SUSY scale:
  The PDG of the particles file of SARAH is used. If this is missing or 0, the mass is not written in order to exclude unphysical states.
- Decay widths and branching ratios for the particle in `ListDecayParticles` and `ListDecayParticles3B`

7.4.12 SPheno.f90

The main program. Calls routines for reading the input, calculating the spectrum, decay widths and low energy observables and writing the output.

7.4.13 Additional Files

- `Makefile`: The number of files for the three body decays is not fixed, therefore the Makefile is generated dynamically each time.
- `LesHouches.in`: Template for an input file of the current model

7.5 Fit to electroweak data

The standard routines for calculating the values of the SM gauge couplings and Yukawa couplings as starting point for the RGE running expect models which contain

- The three SM Yukawa couplings
- The three SM gauge couplings
- Two Higgs doublets giving mass to the SM fermions

In the calculation of the gauge and Yukawa couplings the routines of `SARAH` follow closely the procedure described in ref. [18]: the values for the Yukawa couplings giving mass to the SM fermions and the gauge couplings are determined at the scale M_Z based on the measured values for the quark, lepton and vector boson masses as well as for the gauge couplings. Here, we have included the 1-loop corrections to the mass of W- and Z-boson as well as the SUSY contributions to δ_{VB} for calculating the gauge couplings. Similarly, we have included the complete 1-loop corrections to the self-energies of SM fermions [21]. Moreover, we have resummed the $\tan\beta$ enhanced terms for the calculation of the Yukawa couplings of the b-quark and the τ-lepton as in [18]. The vacuum expectation values v_d and v_u are calculated with respect to the given value of $\tan\beta$ at M_Z.
It is assumed that all SM masses are fixed by these parameters, i.e. the contribution of additional VEVs or the mixing with SUSY particles can't be handled with this routine. This excludes models with R-Parity violation or more generations of SM particles. Also, neutrino observables are not used so far as constraint.

Chapter 8

Numerical values

SARAH offers some basic routine for working with numerical values of parameters and calculating mixing matrices and masses.

8.1 Adding numerical values

There are different ways to define numerical values for a parameter: they can just be added in the parameter file as described in sec. 4.2 or the values from a LesHouches spectrum file can be used. To read automatically a LesHouches file when evaluating the command Start, add

```
SpectrumFile="Name of File"
```

to the model file. The spectrum file must be in the same directory as the model file.
It is also possible to read a spectrum file afterwards by the command

```
ReadSpectrum["Spectrum File"]
```

An additional possibility to add a numerical value during the work with SARAH is

```
SetParameterValue[parameter,value];
```

parameter is the name of the parameter and value the numerical Value. Of course, it is also possible to remove easily a value. For this purpose the command

```
DeleteParameterValue[parameter];
```

is used.

8.2 Calculate Mixing Matrices

After numerical values for all free parameters are defined, the mass eigenstates and entries of mixing matrices can be calculated by

```
CalcMatrices;
```

The eigenvalues and mixing matrices of each mass matrix are saved in two additional variables:

a) Eigenvalues: the eigenvalues of the mass matrix are saved in

```
Mass <> "Name of Eigenstates"
```

b) The numerical value of the mixing matrix are saved in

```
"Name of Mixing Matrix" <> Num
```

Negative Squared Masses It could happen that for some parameter points some eigenvalues of the mass matrices for scalars are getting negative. If this happens, there will appear a warning in the output and the variable `warning` is set to `True`.

Example After using `CalcMatrices`, the numerical value for the down-squark mixing matrix are saved in `MDNum` and the corresponding squark masses are saved in `MassSd`.

8.3 Calculate Numerical Values

To get the numerical value for a term

```
NumericalValue[x];
```

is used. `x` can be e.g. the entry of a mass matrix or a vertex

Examples

a) The numerical value for the Higgs mass matrix is calculated with

```
NumericalValue[MassMatricesFullEWSB[[4]]];
```

b) The numerical value for the interaction between the photon and down squarks is calculated with

```
NumericalValue[Vertex[{VP,Sd,conj[Sd]}][[2,1]]];
```

Appendix A

Gauge anomalies

Before SARAH starts the calculation of the Lagrangian it checks the model for the different triangle anomalies. These anomalies can involve diagrams with three external gauge bosons belonging to the same $U(1)$ or $SU(N)$ gauge group. To be anomaly free the sum over all internal fermions has to vanish

$$U(1)_i^3 \quad : \quad \sum_n {Y_n^i}^3 = 0 \,, \tag{A.1}$$

$$SU(N)_i^3 \quad : \quad \sum_n \mathrm{Tr}(T_n^i T_n^i T_n^i) = 0 \,. \tag{A.2}$$

We label the different gauge groups with the indices i, j, k. Y_n^i is the charge of particle n under the abelian gauge group i while T_n^i is the generator with respect to a non-abelian gauge group.
Combinations of two different gauge groups are possible, if one group is an $U(1)$. Hence, another condition for the absence of anomalies is

$$U(1)_i \times SU(N)_j^2 : \sum_n Y_n^i \,\mathrm{Tr}(T_n^j T_n^j) = 0 \,. \tag{A.3}$$

If more than one $U(1)$ gauge group are present, anomalies can be generated by two or three different $U(1)$ gauge bosons as external fields, too. Therefore, it has to be checked, that

$$U(1)_i \times U(1)_j^2 \quad : \quad \sum_n Y_n^i {Y_n^j}^2 = 0 \,, \tag{A.4}$$

$$U(1)_i \times U(1)_j \times U(1)_k \quad : \quad \sum_n Y_n^i Y_n^j Y_n^k = 0 \tag{A.5}$$

holds. In addition, it has to be checked that there is a even number of $SU(2)$ doublets. This is the necessary for a model in order to be free of the Witten anomaly [24]. If one condition is not fulfilled, a warning is given by SARAH but the model can be evaluated anyway.

Appendix B

Calculation of the Lagrangian of supersymmetric models

We describe in this section the calculation of the complete Lagrangian for a supersymmetric model based on the superpotential and the gauge structure.

Interactions of chiral superfields If we call the superpotential for a given theory W and use ϕ_i for the scalar and ψ_i for the fermionic component of a chiral supermultiplet, the matter interactions can by derived by

$$\mathcal{L}_Y = -\frac{1}{2}W^{ij}\psi_i\psi_j + \text{h.c.}\,, \qquad \mathcal{L}_F = F^{*i}F_i + \text{h.c.} \tag{B.1}$$

with

$$W^{ij} = \frac{\delta^2}{\delta\phi_i\delta\phi_j}W \qquad \text{and} \qquad F^i = -W^{*i} = \frac{\delta W}{\delta\phi_i}\,. \tag{B.2}$$

The first term of eq. (B.1) describes the interaction of two fermions with one scalar, while the second term forms the so called F-terms which describe four-scalar interactions.

Interactions of vector superfields We name the spin-$\frac{1}{2}$ component of a vector supermultiplet λ and the spin-1 component A^μ. The most general Lagrangian only involving these fields is

$$\mathcal{L} = -\frac{1}{4}F^a_{\mu\nu}F^{\mu\nu a} - i\lambda^{\dagger a}\bar{\sigma}^\mu D_\mu\lambda^a \tag{B.3}$$

with the field strength

$$F^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g f^{abc}A^b_\mu A^c_\nu\,, \tag{B.4}$$

and the covariant derivative

$$D_\mu\lambda^a = \partial_\mu\lambda^a + g f^{abc}A^b_\mu\lambda^c\,. \tag{B.5}$$

Here, f^{abc} is the structure constant of the gauge group. Plugging eq. (B.4) in the first term of eq. (B.3) leads to self-interactions of three and four gauge bosons

$$\mathcal{L}_V = -\frac{1}{4}(\partial_\mu A^a_\nu - \partial_\nu A^a_\mu)g f^{abc}A^{\mu,b}A^{\nu,c} - \frac{1}{4}g^2(f_{abc}A^b_\mu A^c_\nu)(f^{ade}A^{\mu,e}A^{\nu,e})\,. \tag{B.6}$$

The second term of eq. (B.3) describes the interactions between vector bosons and gauginos.

Supersymmetric gauge interactions The parts of the Lagrangian with both chiral and vector superfields are the kinetic terms for the fermions and scalars

$$\mathcal{L}_{kin} = -D^\mu \phi^{*i} D_\mu \phi_i - i\psi^{\dagger i} \bar{\sigma}^\mu D_\mu \psi_i \tag{B.7}$$

as well as the interaction between a gaugino and a matter fermion and scalar

$$\mathcal{L}_{GFS} = -\sqrt{2} g(\phi^* T^a \psi)\lambda^a + \text{h.c.} \; . \tag{B.8}$$

Here, T^a are the fundamental generators of the gauge group. Furthermore, the covariant derivatives are

$$D_\mu \phi_i = \partial_\mu \phi_i - igA^a_\mu (T^a \phi)_i \; , \tag{B.9}$$

$$D_\mu \phi^{*i} = \partial_\mu \phi^{*i} + igA^a_\mu (\phi^* T^a)^i \; , \tag{B.10}$$

$$D_\mu \psi_i = \partial_\mu \psi_i - igA^a_\mu (T^a \psi)_i \; , \tag{B.11}$$

In addition, the D-Terms are defined by

$$\mathcal{L}_D = \frac{1}{2} D^a D^a \; . \tag{B.12}$$

The solution of the equations of motion for the auxiliary fields leads to

$$D^a = -g(\phi^* T^a \phi) \; . \tag{B.13}$$

Soft-breaking terms SUSY must be a broken. This can be parametrized by adding soft-breaking terms to the Lagrangian. The possible terms are the mass terms for all scalar matter fields and gauginos

$$\mathcal{L}_{SB} = -m^2_{\phi_i} \phi_i \phi_i^* - \frac{1}{2} M_{\lambda_i} \lambda_i \lambda_i \tag{B.14}$$

as well as soft-breaking interaction corresponding to the superpotential terms

$$\mathcal{L}_{Soft,W} = T\phi_i \phi_j \phi_k + B\phi_i \phi_j + S\phi_i \; . \tag{B.15}$$

Gauge fixing terms and ghost interactions The Lagrangian of a theory without further restrictions is invariant under a general gauge transformation. This invariance leads to severe problems in the quantization of the theory as can be seen in the divergence of functional integrals. Therefore, it is in necessary to add gauge fixing terms to break this gauge invariance.
The general form of the gauge fixing Lagrangian is

$$\mathcal{L}_{GF} = -\frac{1}{2} \sum_a |f(x)^a|^2 \; . \tag{B.16}$$

f_a can be a function of partial derivatives of a gauge boson and a Goldstone boson. The corresponding ghost terms of the ghost fields $\bar{\eta}$ and η are

$$\mathcal{L}_{Ghost} = -\bar{\eta}_a (\delta f^a) \; . \tag{B.17}$$

Here, δ assigns the operator for a BRST transformation. For an unbroken gauge symmetry, the gauge fixing terms in the often chosen R_ξ-gauge are

$$\mathcal{L}_{GF} = -\frac{1}{2R_\xi} \sum_a \left(\partial^\mu V^a_\mu\right)^2 \; . \tag{B.18}$$

Here, V_μ are the gauge boson of the unbroken gauge group. It is often common to choose a distinct value for R_ξ. The most popular gauges are the unitary gauge $R_\xi \to \infty$ and the Feynman-'t Hooft-gauge $R_\xi = 1$. For broken symmetries, the gauge fixings terms are chosen in a way that the mixing terms between vector bosons and scalars disappear from the Lagrangian. Therefore, the common choice for the gauge fixing Lagrangian for theories with the standard model gauge sector after EWSB is

$$\mathcal{L}_{GF,R_\xi} = -\frac{1}{2\xi_\gamma}\left(\partial^\mu \gamma_\mu\right)^2 - \frac{1}{2\xi_Z}\left(\partial^\mu Z_\mu + \xi_Z M_Z G^0\right)^2 + -\frac{1}{\xi_{W^+}}\left(\partial^\mu W_\mu^+ + \xi_{W^+} M_W G^+\right)^2 \ . \tag{B.19}$$

Here, G^0 and G^+ are the Goldstone bosons, which build the longitudinal component of the massive vector bosons.

Appendix C

Parts of the Lagrangian in SARAH

SARAH saves the different parts of the Lagrangian for the different eigenstates in the following variables:

- `LagSV[Name Of States]`: Parts with scalars and vector bosons (= kinetic terms for scalars)
- `LagFFV[Name Of States]`: Parts with fermions and vector bosons (= kinetic terms of scalars)
- `LagSSSS[Name Of States]`: Parts with only scalars (= scalar potential)
- `LagFFS[Name Of States]`: Parts with fermions and scalars
- `LagVVV[Name Of States]`: Parts with three vector bosons
- `LagVVVV[Name Of States]`: Parts with four vector bosons
- `LagGGS[Name Of States]`: Parts with ghosts and scalars
- `LagGGV[Name Of States]`: Parts with ghosts and vector bosons
- `LagSSA[Name Of States]`: Parts with scalars and auxiliary fields (only needed for `CalcHep` output)

In addition, for an effective theory, there might exist

- `LagSSSSSS[Name Of States]`: Dimension 6 operators with only scalars
- `LagSSSVVV[Name Of States]`: Dimension 6 operators with scalars and fermions
- `LagFFFF[Name Of States]`: Dimension 6 operators with only fermions
- `LagFFSS[Name Of States]`: Dimension 5 operators with fermions and scalars
- `LagFFVV[Name Of States]`: Dimension 6 operators with fermions and vector bosons

Moreover, the different results of the calculation of the Lagrangian in gauge eigenstates are also saved separately. The variable names are:

- Superpotential: `Superpotential`
- Fermion - scalar interactions coming from the superpotential: `Wij`
- F-Terms: `FTerms`

- Scalar soft-breaking masses: `SoftScalarMass`
- Gaugino masses: `SoftGauginoMass`
- Soft-breaking couplings: `SoftW`
- Kinetic terms for scalars: `KinScalar`
- Kinetic terms for fermions: `KinFermion`
- D-Terms: `DTerms`
- Interactions between gauginos and a scalar and a fermion: `FSGaugino`
- Trilinear self-interactions of gauge bosons: `GaugeTri`
- Quartic self-interactions of gauge bosons: `GaugeQuad`
- Interactions between vector bosons and gauginos: `BosonGaugino`

Furthermore, the additional interactions and the redefinition of existing interactions are saved in `LagRedefinition`.

Appendix D

Calculation of Group Factors

SARAH supports not only chiral superfields in the fundamental representation but in any irreducible representation of $SU(N)$. In most cases, it is possible to fix the transformation properties of the chiral superfield by stating the corresponding dimension D. If the dimension is not unique, also the Dynkin labels are needed. For calculating kinetic terms and D-terms, it is necessary to derive from representation the corresponding generators. Furthermore, the eigenvalues C_2 of the quadratic Casimir for any irreducible representation r

$$T^a T^a \phi(r) = C_2(r)\phi(r) \tag{D.1}$$

as well as the Dynkin index I

$$Tr(T^a T^b)\phi(r) = I\delta_{ab}\phi(r) \tag{D.2}$$

are needed for the calculation of the RGEs. All of that is derived by `SARAH` due to the technique of Young tableaux. The following steps are evolved:

a) The corresponding Young tableaux fitting to the dimension D is calculated using the hook formula:

$$D = \Pi_i \frac{N + d_i}{h_i} \tag{D.3}$$

d_i is the distance of the i. box to the left upper corner and h_i is the hook of that box.

b) The vector for the highest weight Λ in Dynkin basis is extracted from the tableaux.

c) The fundamental weights for the given gauge group are calculated.

d) The value of $C_2(r)$ is calculated using the Weyl formula

$$C_2(r) = (\Lambda, \Lambda + \rho)\ . \tag{D.4}$$

ρ is the Weyl vector.

e) The Dynkin index $I(r)$ is calculated from $C_2(r)$. For this step, the value for the fundamental representation is normalized to $\frac{1}{2}$.

$$I(r) = C_2(r)\frac{D(r)}{D(G)} \tag{D.5}$$

With $D(G)$ as dimension of the adjoint representation.

f) The number of co- and contra-variant indices is extracted from the Young tableaux. With this information, the generators are written as tensor product.

The user can calculate this information independently from the model using the new command

```
CheckIrrepSUN[Dim,N]
```

`Dim` is the dimension of the irreducible representation and `N` is the dimension of the $SU(N)$ gauge group. The result is a vector containing the following information: (i) repeating the dimension of the field, (ii) number of covariant indices, (iii) number of contravariant indices, (iv) value of the quadratic Casimir $C_2(r)$, (v) value of the Dynkin index $I(r)$, (vi) Dynkin labels for the highest weight.

Examples

a) Fundamental representation The properties of a particle, transforming under the fundamental representation of $SU(3)$ are calculated via `CheckIrrepSUN[3,3]`. The output is the well known result

```
{3, 1, 0, 4/3, 1/2, {1, 0}}
```

b) Adjoint representation The properties of a field transforming as **24** of $SU(5)$ are calculated by `CheckIrrepSUN[24,5]` . The output will be

```
{24, 1, 1, 5, 5, {1, 0, 0, 1}}
```

c) Different representations with the same dimension The **70** under $SU(5)$ is not unique. Therefore, `CheckIrrepSUN[{70, {0, 0, 0, 4}}, 5]` returns

```
{70, 0, 4, 72/5, 42, {0, 0, 0, 4}}
```

while `CheckIrrepSUN[{70, {2, 0, 0, 1}}, 5]` leads to

```
{70, 2, 1, 42/5, 49/2, {2, 0, 0, 1}}
```

Appendix E

Conventions and generic expressions

E.1 Renormalization group equations

We summarize in this section the used equations for the calculation of the one- and two-loop RGEs in SARAH. These equations are extensively discussed in literature, see e.g. [20, 25, 26, 27, 28, 29, 30, 31, 25, 32]. For a general $N = 1$ supersymmetric gauge theory with superpotential

$$W(\phi) = L_i\phi_i + \frac{1}{2}\mu^{ij}\phi_i\phi_j + \frac{1}{6}Y^{ijk}\phi_i\phi_j\phi_k \ , \tag{E.1}$$

the soft SUSY-breaking scalar terms are given by

$$V_{\text{soft}} = \left(S^i\phi_i + \frac{1}{2}b^{ij}\phi_i\phi_j + \frac{1}{6}h^{ijk}\phi_i\phi_j\phi_k + \text{c.c.}\right) + (m^2)^i{}_j\phi_i\phi_j^* + \frac{1}{2}M_\lambda\lambda_a\lambda_a \ . \tag{E.2}$$

The anomalous dimensions are given by

$$\gamma_i^{(1)j} = \frac{1}{2}Y_{ipq}Y^{jpq} - 2\delta_i^j g^2 C(i) \ , \tag{E.3}$$

$$\begin{aligned}\gamma_i^{(2)j} =& -\frac{1}{2}Y_{imn}Y^{npq}Y_{pqr}Y^{mrj} + g^2 Y_{ipq}Y^{jpq}[2C(p) - C(i)] \\ &+ 2\delta_i^j g^4[C(i)S(R) + 2C(i)^2 - 3C(G)C(i)] \ ,\end{aligned} \tag{E.4}$$

and the β-functions for the gauge couplings are given by

$$\beta_g^{(1)} = g^3\left[S(R) - 3C(G)\right] \ , \tag{E.5}$$

$$\beta_g^{(2)} = g^5\left\{-6[C(G)]^2 + 2C(G)S(R) + 4S(R)C(R)\right\} - g^3 Y^{ijk}Y_{ijk}C(k)/D(G) \ . \tag{E.6}$$

Here, $C(i)$ is the quadratic Casimir for a specific superfield and $C(R), C(G)$ are the quadratic Casimirs for the matter and adjoint representations, respectively. $D(G)$ is the dimension of the adjoint representation. The β-functions for the superpotential parameters can be obtained by using superfield technique. The obtained expressions are

$$\beta_Y^{ijk} = Y^{ijp}\left[\frac{1}{16\pi^2}\gamma_p^{(1)k} + \frac{1}{(16\pi^2)^2}\gamma_p^{(2)k}\right] + (k \leftrightarrow i) + (k \leftrightarrow j) \ , \tag{E.7}$$

$$\beta_\mu^{ij} = \mu^{ip}\left[\frac{1}{16\pi^2}\gamma_p^{(1)j} + \frac{1}{(16\pi^2)^2}\gamma_p^{(2)j}\right] + (j \leftrightarrow i) \ , \tag{E.8}$$

$$\beta_L^i = L^p\left[\frac{1}{16\pi^2}\gamma_p^{(1)i} + \frac{1}{(16\pi^2)^2}\gamma_p^{(2)i}\right] \ . \tag{E.9}$$

The expressions for trilinear, soft-breaking terms are

$$\frac{d}{dt}h^{ijk} = \frac{1}{16\pi^2}\left[\beta_h^{(1)}\right]^{ijk} + \frac{1}{(16\pi^2)^2}\left[\beta_h^{(2)}\right]^{ijk} \ , \tag{E.10}$$

with

$$\begin{aligned}\left[\beta_h^{(1)}\right]^{ijk} =& \frac{1}{2}h^{ijl}Y_{lmn}Y^{mnk} + Y^{ijl}Y_{lmn}h^{mnk} - 2\left(h^{ijk} - 2MY^{ijk}\right)g^2C(k) \\ &+ (k \leftrightarrow i) + (k \leftrightarrow j) \ , \end{aligned} \tag{E.11}$$

$$\begin{aligned}\left[\beta_h^{(2)}\right]^{ijk} =& -\frac{1}{2}h^{ijl}Y_{lmn}Y^{npq}Y_{pqr}Y^{mrk} \\ &- Y^{ijl}Y_{lmn}Y^{npq}Y_{pqr}h^{mrk} - Y^{ijl}Y_{lmn}h^{npq}Y_{pqr}Y^{mrk} \\ &+ \left(h^{ijl}Y_{lpq}Y^{pqk} + 2Y^{ijl}Y_{lpq}h^{pqk} - 2MY^{ijl}Y_{lpq}Y^{pqk}\right)g^2\left[2C(p) - C(k)\right] \\ &+ \left(2h^{ijk} - 8MY^{ijk}\right)g^4\left[C(k)S(R) + 2C(k)^2 - 3C(G)C(k)\right] \\ &+ (k \leftrightarrow i) + (k \leftrightarrow j) \ . \end{aligned} \tag{E.12}$$

For the bilinear soft-breaking parameters, the expressions read

$$\frac{d}{dt}b^{ij} = \frac{1}{16\pi^2}\left[\beta_b^{(1)}\right]^{ij} + \frac{1}{(16\pi^2)^2}\left[\beta_b^{(2)}\right]^{ij} \ , \tag{E.13}$$

with

$$\begin{aligned}\left[\beta_b^{(1)}\right]^{ij} =& \frac{1}{2}b^{il}Y_{lmn}Y^{mnj} + \frac{1}{2}Y^{ijl}Y_{lmn}b^{mn} + \mu^{il}Y_{lmn}h^{mnj} - 2\left(b^{ij} - 2M\mu^{ij}\right)g^2C(i) \\ &+ (i \leftrightarrow j) \ , \end{aligned} \tag{E.14}$$

$$\begin{aligned}\left[\beta_b^{(2)}\right]^{ij} =& -\frac{1}{2}b^{il}Y_{lmn}Y^{pqn}Y_{pqr}Y^{mrj} - \frac{1}{2}Y^{ijl}Y_{lmn}b^{mr}Y_{pqr}Y^{pqn} \\ &- \frac{1}{2}Y^{ijl}Y_{lmn}\mu^{mr}Y_{pqr}h^{pqn} - \mu^{il}Y_{lmn}h^{npq}Y_{pqr}Y^{mrj} \\ &- \mu^{il}Y_{lmn}Y^{npq}Y_{pqr}h^{mrj} + 2Y^{ijl}Y_{lpq}\left(b^{pq} - \mu^{pq}M\right)g^2C(p) \\ &+ \left(b^{il}Y_{lpq}Y^{pqj} + 2\mu^{il}Y_{lpq}h^{pqj} - 2\mu^{il}Y_{lpq}Y^{pqj}M\right)g^2\left[2C(p) - C(i)\right] \\ &+ \left(2b^{ij} - 8\mu^{ij}M\right)g^4\left[C(i)S(R) + 2C(i)^2 - 3C(G)C(i)\right] \\ &+ (i \leftrightarrow j) \ , \end{aligned} \tag{E.15}$$

Finally, the RGEs for the linear soft-breaking parameters are

$$\frac{d}{dt}S^i = \frac{1}{16\pi^2}\left[\beta_S^{(1)}\right]^i + \frac{1}{(16\pi^2)^2}\left[\beta_S^{(2)}\right]^i \ , \tag{E.16}$$

with

$$\left[\beta_S^{(1)}\right]^i = \frac{1}{2}Y^{iln}Y_{pln}S^p + L^pY_{pln}h^{iln} + \mu^{ik}Y_{kln}B^{ln} + 2Y^{ikp}(m^2)^l_p\mu_{kl} + h^{ikl}B_{kl} \ , \tag{E.17}$$

$$\begin{aligned}\left[\beta_S^{(2)}\right]^i =& 2g^2C(l)Y^{ikl}Y_{pkl}S^p - \frac{1}{2}Y^{ikq}Y_{qst}Y^{lst}Y_{pkl}S^p - 4g^2C(l)(Y^{ikl}M - h^{ikl})Y_{pkl}L^p \\ &- \left[Y^{ikq}Y_{qst}h^{lst}Y_{pkl} + h^{ikq}Y_{qst}Y^{lst}Y_{pkl}\right]L^p - 4g^2C(l)Y_{jnl}(\mu^{nl}M - B^{nl})\mu^{ij} \\ &- \left[Y_{jnq}h^{qst}Y_{lst}\mu^{nl} + Y_{jnq}Y^{qst}Y_{lst}B^{nl}\right]\mu^{ij} + 4g^2C(l)(2Y^{ikl}\mu_{kl}|M|^2 - Y^{ikl}B_{kl}M \\ &- h^{ikl}\mu_{kl}M^* + h^{ikl}B_{kl} + Y^{ipl}(m^2)^k_p\mu_{kl} + Y^{ikp}(m^2)^l_p\mu_{kl}) \\ &- \Big[Y^{ikq}Y_{qst}h^{lst}B_{kl} + h^{ikq}Y_{qst}Y^{lst}B_{kl} + Y^{ikq}h_{qst}h^{lst}\mu_{kl} + h^{ikq}h_{qst}Y^{lst}\mu_{kl} \\ &+ Y^{ipq}(m^2)^k_pY_{qst}Y^{lst}\mu_{kl} + Y^{ikq}Y_{qst}Y^{pst}(m^2)^l_p\mu_{kl} + Y^{ikp}(m^2)^q_pY_{qst}Y^{lst}\mu_{kl} \\ &+ 2Y^{ikq}Y_{qsp}(m^2)^p_tY^{lst}\mu_{kl}\Big] \ .\end{aligned} \tag{E.18}$$

With this results, the list of the β-functions for all couplings is complete. Now, we turn to the RGEs for the gaugino masses, squared masses of scalars and vacuum expectation values. The result for the gaugino masses is

$$\frac{d}{dt}M = \frac{1}{16\pi^2}\beta_M^{(1)} + \frac{1}{(16\pi^2)^2}\beta_M^{(2)} \ , \tag{E.19}$$

with

$$\beta_M^{(1)} = g^2\left[2S(R) - 6C(G)\right]M \ , \tag{E.20}$$

$$\begin{aligned}\beta_M^{(2)} =& g^4\left\{-24[C(G)]^2 + 8C(G)S(R) + 16S(R)C(R)\right\}M \\ &+ 2g^2\left[h^{ijk} - MY^{ijk}\right]Y_{ijk}C(k)/D(G) \ .\end{aligned} \tag{E.21}$$

The one- and two-loop RGEs for the scalar mass parameters read

$$\frac{d}{dt}m_{ij} = \frac{1}{16\pi^2}\left[\beta_{m^2}^{(1)}\right]^j_i + \frac{1}{(16\pi^2)^2}\left[\beta_{m^2}^{(2)}\right]^j_i \ , \tag{E.22}$$

$$\tag{E.23}$$

with

$$\begin{aligned}
\left[\beta^{(1)}_{m^2}\right]^j_i =& \frac{1}{2}Y_{ipq}Y^{pqn}(m^2)^j_n + \frac{1}{2}Y^{jpq}Y_{pqn}(m^2)^n_i + 2Y_{ipq}Y^{jpr}(m^2)^q_r \\
&+ h_{ipq}h^{jpq} - 8\delta^j_i MM^\dagger g^2 C(i) + 2g^2\mathbf{t}^{Aj}_i \mathrm{Tr}[\mathbf{t}^A m^2] \,, \qquad \text{(E.24)} \\
\left[\beta^{(2)}_{m^2}\right]^j_i =& -\frac{1}{2}(m^2)^l_i Y_{lmn}Y^{mrj}Y_{pqr}Y^{pqn} - \frac{1}{2}(m^2)^j_l Y^{lmn}Y_{mri}Y^{pqr}Y_{pqn} \\
&- Y_{ilm}Y^{jnm}(m^2)^l_r Y_{npq}Y^{rpq} - Y_{ilm}Y^{jnm}(m^2)^r_n Y_{rpq}Y^{lpq} \\
&- Y_{ilm}Y^{jnr}(m^2)^l_n Y_{pqr}Y^{pqm} - 2Y_{ilm}Y^{jln}Y_{npq}Y^{mpr}(m^2)^q_r \\
&- Y_{ilm}Y^{jln}h_{npq}h^{mpq} - h_{ilm}h^{jln}Y_{npq}Y^{mpq} \\
&- h_{ilm}Y^{jln}Y_{npq}h^{mpq} - Y_{ilm}h^{jln}h_{npq}Y^{mpq} \\
&+ \Big[(m^2)^l_i Y_{lpq}Y^{jpq} + Y_{ipq}Y^{lpq}(m^2)^j_l + 4Y_{ipq}Y^{jpl}(m^2)^q_l + 2h_{ipq}h^{jpq} \\
&- 2h_{ipq}Y^{jpq}M - 2Y_{ipq}h^{jpq}M^\dagger + 4Y_{ipq}Y^{jpq}MM^\dagger\Big] g^2\left[C(p) + C(q) - C(i)\right] \\
&- 2g^2\mathbf{t}^{Aj}_i(\mathbf{t}^A m^2)^l_r Y_{lpq}Y^{rpq} + 8g^4\mathbf{t}^{Aj}_i \mathrm{Tr}[\mathbf{t}^A C(r) m^2] \\
&+ \delta^j_i g^4 MM^\dagger\left[24C(i)S(R) + 48C(i)^2 - 72C(G)C(i)\right] \\
&+ 8\delta^j_i g^4 C(i)(\mathrm{Tr}[S(r)m^2] - C(G)MM^\dagger) \,. \qquad \text{(E.25)}
\end{aligned}$$

The RGEs for a VEV v^i is proportional to the anomalous dimension of the chiral superfield whose scalar component receives the VEV

$$\frac{d}{dt}v^i = v^p\left[\frac{1}{16\pi^2}\gamma^{(1)i}_p + \frac{1}{(16\pi^2)^2}\gamma^{(2)i}_p\right] \qquad \text{(E.26)}$$

E.2 One-loop amplitudes for one- and two-point functions

We used for the calculation of the one-loop self energies and the one-loop corrections to the tadpoles in $\overline{\mathrm{DR}}$-scheme the scalar functions defined in [21]. The basis integrals are

$$A_0(m) = 16\pi^2 Q^{4-n}\int\frac{d^nq}{i\,(2\pi)^n}\frac{1}{q^2 - m^2 + i\varepsilon} \,, \qquad \text{(E.27)}$$

$$B_0(p,m_1,m_2) = 16\pi^2 Q^{4-n}\int\frac{d^nq}{i\,(2\pi)^n}\frac{1}{\left[q^2 - m_1^2 + i\varepsilon\right]\left[(q-p)^2 - m_2^2 + i\varepsilon\right]} \,, \qquad \text{(E.28)}$$

with the renormalization scale Q. The integrals are regularized by integrating in $n = 4 - 2\epsilon$ dimensions. The result for A_0 is

$$A_0(m) = m^2\left(\frac{1}{\hat{\epsilon}} + 1 - \ln\frac{m^2}{Q^2}\right) \,, \qquad \text{(E.29)}$$

where $1/\hat{\epsilon} = 1/\epsilon - \gamma_E + \ln 4\pi$. The function B_0 has the analytic expression

$$B_0(p,m_1,m_2) = \frac{1}{\hat{\epsilon}} - \ln\left(\frac{p^2}{Q^2}\right) - f_B(x_+) - f_B(x_-) \,, \qquad \text{(E.30)}$$

with

$$x_\pm = \frac{s \pm \sqrt{s^2 - 4p^2(m_1^2 - i\varepsilon)}}{2p^2}, \qquad f_B(x) = \ln(1-x) - x\ln(1-x^{-1}) - 1, \tag{E.31}$$

and $s = p^2 - m_2^2 + m_1^2$. All the other, necessary functions can be expressed by A_0 and B_0. For instance,

$$B_1(p, m_1, m_2) = \frac{1}{2p^2}\left[A_0(m_2) - A_0(m_1) + (p^2 + m_1^2 - m_2^2)B_0(p, m_1, m_2)\right], \tag{E.32}$$

and

$$\begin{aligned} B_{22}(p, m_1, m_2) = \frac{1}{6}\Bigg\{ & \frac{1}{2}\left(A_0(m_1) + A_0(m_2)\right) + \left(m_1^2 + m_2^2 - \frac{1}{2}p^2\right) B_0(p, m_1, m_2) \\ & + \frac{m_2^2 - m_1^2}{2p^2}\left[A_0(m_2) - A_0(m_1) - (m_2^2 - m_1^2)B_0(p, m_1, m_2)\right] \\ & + m_1^2 + m_2^2 - \frac{1}{3}p^2 \Bigg\}. \end{aligned} \tag{E.33}$$

Furthermore, for the vector boson self-energies it is useful to define

$$F_0(p, m_1, m_2) = A_0(m_1) - 2A_0(m_2) - (2p^2 + 2m_1^2 - m_2^2)B_0(p, m_1, m_2), \tag{E.34}$$

$$G_0(p, m_1, m_2) = (p^2 - m_1^2 - m_2^2)B_0(p, m_1, m_2) - A_0(m_1) - A_0(m_2), \tag{E.35}$$

$$H_0(p, m_1, m_2) = 4B_{22}(p, m_1, m_2) + G(p, m_1, m_2), \tag{E.36}$$

$$\tilde{B}_{22}(p, m_1, m_2) = B_{22}(p, m_1, m_2) - \frac{1}{4}A_0(m_1) - \frac{1}{4}A_0(m_2) \tag{E.37}$$

In all calculations, several coefficient are involved:

- c_S is the symmetry factor: if the particles in the loop are indistinguishable, the weight of the contribution is only half of the weight in the case of distinguishable particles. If two different charge flows are possible in the loop, the weight of the diagram is doubled.
- c_C is a charge factor: for corrections due to vector bosons in the adjoint representation this is the Casimir of the corresponding group. For corrections due to matter fields this can be, for instance, a color factor for quarks/squarks. For corrections of vector bosons in the adjoint representation this is normally the Dynkin index of the gauge group.
- c_R is 2 for real fields and Majorana fermions in the loop and 1 otherwise.

We use in the following Γ for non-chiral interactions and Γ_L/Γ_R for chiral interactions. If two vertices are involved, the interaction of the incoming particle has an upper index 1 and for the outgoing field an upper index 2 is used.

E.2.1 One-loop tadpoles

a) Fermion loop (generic name in `SARAH`: `FFS`):

$$T = 8 c_S c_C m_F \Gamma A_0(m_F^2) \tag{E.38}$$

b) Scalar loop (generic name in SARAH: SSS):

$$T = -2c_S c_C \Gamma A_0(m_S^2) \tag{E.39}$$

c) Vector boson loop (generic name in SARAH: SVV):

$$T = 6c_S c_C \Gamma A_0(m_V^2) \tag{E.40}$$

E.2.2 One-loop self-energies

E.2.2.0.1 Corrections to fermion

a) Fermion-scalar loop (generic name in SARAH: FFS):

$$\begin{aligned}
\Sigma^S(p^2) &= m_F c_S c_C c_R \Gamma_R^1 \Gamma_L^{2,*} B_0(p^2, m_F^2, m_S^2) \\
\Sigma^R(p^2) &= -c_S c_C c_R \frac{1}{2} \Gamma_R^1 \Gamma_R^{2,*} B_1(p^2, m_F^2, m_S^2) \\
\Sigma^L(p^2) &= -c_S c_C c_R \frac{1}{2} \Gamma_L^1 \Gamma_L^{2,*} B_1(p^2, m_F^2, m_S^2)
\end{aligned}$$

b) Fermion-vector boson loop (generic name in SARAH: FFV):

$$\begin{aligned}
\Sigma^S(p^2) &= -4c_S c_C c_R m_F \Gamma_L^1 \Gamma_R^{2,*} B_0(p^2, m_F^2, m_S^2) \\
\Sigma^R(p^2) &= -c_S c_C c_R \Gamma_L^1 \Gamma_L^{2,*} B_1(p^2, m_F^2, m_S^2) \\
\Sigma^L(p^2) &= -c_S c_C c_R \Gamma_R^1 \Gamma_R^{2,*} B_1(p^2, m_F^2, m_S^2)
\end{aligned}$$

E.2.2.0.2 Corrections to scalar

a) Fermion loop (generic name in SARAH: FFS):

$$\Pi(p^2) = c_S c_C c_R \left((\Gamma_L^1 \Gamma_L^{2,*} + \Gamma_R^1 \Gamma_R^{2,*}) G_0(p^2, m_F^2, m_S^2) + (\Gamma_L^1 \Gamma_R^{2,*} + \Gamma_R^1 \Gamma_L^{2,*}) B_0(p^2, m_F^2, m_S^2) \right) \tag{E.41}$$

b) Scalar loop (two 3-point interactions, generic name in SARAH: SSS):

$$\Pi(p^2) = c_S c_C c_R \Gamma^1 \Gamma^{2,*} B_0(p^2, m_F^2, m_S^2) \tag{E.42}$$

c) Scalar loop (4-point interaction, generic name in SARAH: SSSS):

$$\Pi(p^2) = -c_S c_C \Gamma A_0(m_S^2) \tag{E.43}$$

d) Vector boson-scalar loop (generic name in SARAH: SSV):

$$\Pi(p^2) = c_S c_C c_R \Gamma^1 \Gamma^{2,*} F_0(p^2, m_F^2, m_S^2) \tag{E.44}$$

e) Vector boson loop (two 3-point interactions, generic name in SARAH: SVV):

$$\Pi(p^2) = c_S c_C c_R \frac{7}{2} \Gamma^1 \Gamma^{2,*} B_0(p^2, m_F^2, m_S^2) \tag{E.45}$$

f) Vector boson loop (4-point interaction, generic name in SARAH: SSVV):

$$\Pi(p^2) = c_S c_C \Gamma A_0(m_V^2) \tag{E.46}$$

E.2.2.0.3 Corrections to vector boson

a) Fermion loop (generic name in SARAH: FFV):

$$\Pi^T(p^2) = c_S c_C c_R \left((|\Gamma^1_L|^2 + |\Gamma^1_R|^2)H_0(p^2, m_V^2, m_F^2) + 4Re(\Gamma^1_L \Gamma^2_R) B_0(p^2, m_V^2, m_F^2)\right) \tag{E.47}$$

b) Scalar loop (generic name in SARAH: SSV):

$$\Pi^T(p^2) = -4 c_S c_C c_R |\Gamma|^2 B_{22}(p^2, m_{S_1}^2, m_{S_2}^2) \tag{E.48}$$

c) Vector boson loop (generic name in SARAH: VVV):

$$\Pi^T(p^2) = |\Gamma|^2 c_S c_C c_R \left(-(4p^2 + m_{V_1}^2 + m_{V_2}^2)B_0(p^2, m_{V_1}^2, m_{V_1}^2) - 8B_{22}(p^2, m_{S_1}^2, m_{S_2}^2)\right) \tag{E.49}$$

d) Vector-Scalar-Loop (generic name in SARAH: SVV):

$$\Pi^T(p^2) = |\Gamma|^2 c_S c_C c_R B_0(p^2, m_V^2, m_S^2) \tag{E.50}$$

We need here only the diagrams involving three point interactions because the 4-point interactions are related to them due to gauge invariance.

E.2.3 One-loop corrections to masses

The one-loop self-energies can be used to calculate the one-loop masses and mass matrices. For the one-loop corrections of scalars, the radiatively corrected mass matrix is

$$m_{1L}^{2,S}(p_i^2) = m_T^{2,S} - \Pi_{SS}(p_i^2), \tag{E.51}$$

while the one-loop mass of a vector boson V is given by

$$m_{1L}^{2,V}(Q) = m_T^{2,V} + \mathrm{Re}\{\Pi^T_{VV}(m_T^{2,V})\}. \tag{E.52}$$

According to the conventions of the counter terms of [21], the one-loop mass matrices $M_{1L}^{\tilde{\chi}^0}$ of Majorana fermions are connected to the one-loop self-energies and tree-level mass matrix $M_T^{\tilde{\chi}^0}$ by

$$\begin{aligned} M_{1L}^{\tilde{\chi}^0}(p_i^2) \;=\; & M_T^{\tilde{\chi}^0} - \frac{1}{2}\Big[\Sigma_S^0(p_i^2) + \Sigma_S^{0,T}(p_i^2) + \left(\Sigma_L^{0,T}(p_i^2) + \Sigma_R^0(p_i^2)\right) M_T^{\tilde{\chi}^0} \\ & + M_T^{\tilde{\chi}^0}\left(\Sigma_R^{0,T}(p_i^2) + \Sigma_L^0(p_i^2)\right)\Big]. \end{aligned} \tag{E.53}$$

In the case of Dirac fermions, the one-loop corrected mass matrix is

$$M_{1L}^{\tilde{\chi}^+}(p_i^2) = M_T^{\tilde{\chi}^+} - \Sigma_S^+(p_i^2) - \Sigma_R^+(p_i^2) M_T^{\tilde{\chi}^+} - M_T^{\tilde{\chi}^+}\Sigma_L^+(p_i^2). \tag{E.54}$$

Appendix F

The minimal supersymmetric standard model

F.1 Vector Superfields

SF	Spin $\frac{1}{2}$	Spin 1	$SU(N)$	Coupling	Name
$\hat{B}$	$\lambda_{\tilde{B}}$	B	$U(1)$	g_1	hypercharge
$\hat{W}$	$\lambda_{\tilde{W}}$	W	SU(2)	g_2	left
$\hat{g}$	$\lambda_{\tilde{g}}$	g	SU(3)	g_3	color

F.2 Chiral Superfields

SF	Spin 0	Spin $\frac{1}{2}$	Generations	$(U(1) \otimes \mathrm{SU}(2) \otimes \mathrm{SU}(3))$
$\hat{q}$	$\tilde{q}$	q	3	$(\frac{1}{6}, \mathbf{2}, \mathbf{3})$
$\hat{l}$	$\tilde{l}$	l	3	$(-\frac{1}{2}, \mathbf{2}, \mathbf{1})$
$\hat{H}_d$	H_d	$\tilde{H}_d$	1	$(-\frac{1}{2}, \mathbf{2}, \mathbf{1})$
$\hat{H}_u$	H_u	$\tilde{H}_u$	1	$(\frac{1}{2}, \mathbf{2}, \mathbf{1})$
$\hat{d}$	$\tilde{d}_R^*$	d_R^*	3	$(\frac{1}{3}, \mathbf{1}, \overline{\mathbf{3}})$
$\hat{u}$	$\tilde{u}_R^*$	u_R^*	3	$(-\frac{2}{3}, \mathbf{1}, \overline{\mathbf{3}})$
$\hat{e}$	$\tilde{e}_R^*$	e_R^*	3	$(1, \mathbf{1}, \mathbf{1})$

F.3 Superpotential and Lagrangian

F.3.0.0.4 Superpotential

$$W = Y_u\, \hat{u}\, \hat{q}\, \hat{H}_u \; - Y_d\, \hat{d}\, \hat{q}\, \hat{H}_d \; - Y_e\, \hat{e}\, \hat{l}\, \hat{H}_d \; + \mu\, \hat{H}_u\, \hat{H}_d \tag{F.1}$$

F.3.0.0.5 Softbreaking terms

$$\begin{aligned}-L_{SB,W} = &- H_d^0 H_u^0 B_\mu + H_d^- H_u^+ B_\mu + H_d^0 \tilde{d}^*_{R,i\alpha}\delta_{\alpha\beta}\tilde{d}_{L,j\beta}T_{d,ij} - H_d^- \tilde{d}^*_{R,i\alpha}\delta_{\alpha\beta}\tilde{u}_{L,j\beta}T_{d,ij} \\ &+ H_d^0 \tilde{e}^*_{R,i}\tilde{e}_{L,j}T_{e,ij} - H_d^- \tilde{e}^*_{R,i}\tilde{\nu}_{L,j}T_{e,ij} - H_u^+ \tilde{u}^*_{R,i\alpha}\delta_{\alpha\beta}\tilde{d}_{L,j\beta}T_{u,ij} + H_u^0 \tilde{u}^*_{R,i\alpha}\delta_{\alpha\beta}\tilde{u}_{L,j\beta}T_{u,ij} + \text{h.c.}\end{aligned} \tag{F.2}$$

$$\begin{aligned}-L_{SB,\phi} = &+ m^2_{H_d}|H_d^0|^2 + m^2_{H_d}|H_d^-|^2 + m^2_{H_u}|H_u^0|^2 + m^2_{H_u}|H_u^+|^2 + \tilde{d}^*_{L,j\beta}\delta_{\alpha\beta}m^2_{q,ij}\tilde{d}_{L,i\alpha} \\ &+ \tilde{d}^*_{R,i\alpha}\delta_{\alpha\beta}m^2_{d,ij}\tilde{d}_{R,j\beta} + \tilde{e}^*_{L,j}m^2_{l,ij}\tilde{e}_{L,i} + \tilde{e}^*_{R,i}m^2_{e,ij}\tilde{e}_{R,j} + \tilde{u}^*_{L,j\beta}\delta_{\alpha\beta}m^2_{q,ij}\tilde{u}_{L,i\alpha} \\ &+ \tilde{u}^*_{R,i\alpha}\delta_{\alpha\beta}m^2_{u,ij}\tilde{u}_{R,j\beta} + \tilde{\nu}^*_{L,j}m^2_{l,ij}\tilde{\nu}_{L,i}\end{aligned} \tag{F.3}$$

$$-L_{SB,\lambda} = \frac{1}{2}\left(\lambda^2_{\tilde{B}}M_1 + M_2\lambda^2_{\tilde{W},i} + M_3\lambda^2_{\tilde{g},i} + \text{h.c.}\right) \tag{F.4}$$

F.3.1 Gauge fixing terms

F.3.1.0.6 Gauge fixing terms for gauge eigenstates

$$L_{GF} = -\frac{1}{2\xi_G}\partial_\mu g_\alpha - \frac{1}{2\xi_W}\partial_\mu W^i \tag{F.5}$$

F.3.1.0.7 Gauge fixing terms for mass eigenstates after EWSB

$$L_{GF} = -\frac{1}{2\xi_P}\partial_\mu\gamma - \frac{1}{2\xi_G}\partial_\mu g_\alpha - \frac{1}{2\xi_Z}\Big(-A_1^0 m_Z\xi_Z + \partial_\mu Z\Big) - \frac{1}{\xi_W}\Big(iH_1^- m_{W^-}\xi_W + \partial_\mu W^-\Big) \tag{F.6}$$

F.4 Vacuum expectation values

$$H_d^0 = \frac{1}{\sqrt{2}}\left(\phi_d + i\sigma_d + v_d\right), \qquad H_u^0 = \frac{1}{\sqrt{2}}\left(\phi_u + i\sigma_u + v_u\right) \tag{F.7}$$

F.5 Rotations of vector bosons and gauginos after EWSB

$$W^-_{1\rho} = \frac{1}{\sqrt{2}}W^-_\rho + \frac{1}{\sqrt{2}}W^+_\rho, \qquad W^-_{2\rho} = -i\frac{1}{\sqrt{2}}W^-_\rho + i\frac{1}{\sqrt{2}}W^+_\rho \tag{F.8}$$

$$W^-_{3\rho} = \cos\Theta_W Z_\rho + \sin\Theta_W\gamma_\rho, \qquad B_\rho = \cos\Theta_W\gamma_\rho - \sin\Theta_W Z_\rho \tag{F.9}$$

$$\lambda_{\tilde{W},1} = \frac{1}{\sqrt{2}}\tilde{W}^- + \frac{1}{\sqrt{2}}\tilde{W}^+, \qquad \lambda_{\tilde{W},2} = -i\frac{1}{\sqrt{2}}\tilde{W}^- + i\frac{1}{\sqrt{2}}\tilde{W}^+, \qquad \lambda_{\tilde{W},3} = \tilde{W}^0 \tag{F.10}$$

F.6 Rotations in matter sector to mass eigenstates after EWSB

In the following, Greek letters α_i, β_i refer to color indices and o_i, p_i to generations indices.

a) Mass matrix for neutralinos, basis: $\left(\lambda_{\tilde{B}}, \tilde{W}^0, \tilde{H}_d^0, \tilde{H}_u^0\right)$

$$m_{\tilde{\chi}^0} = \begin{pmatrix} M_1 & 0 & -\frac{1}{2}g_1 v_d & \frac{1}{2}g_1 v_u \\ 0 & M_2 & \frac{1}{2}g_2 v_d & -\frac{1}{2}g_2 v_u \\ -\frac{1}{2}g_1 v_d & \frac{1}{2}g_2 v_d & 0 & -\mu \\ \frac{1}{2}g_1 v_u & -\frac{1}{2}g_2 v_u & -\mu & 0 \end{pmatrix} \tag{F.11}$$

This matrix is diagonalized by N:

$$N m_{\tilde{\chi}^0} N^\dagger = m_{\tilde{\chi}^0}^{dia} \tag{F.12}$$

with

$$\lambda_{\tilde{B}} = \sum_{t_2} N^*_{j1}\lambda^0_j\,, \qquad \tilde{W}^0 = \sum_{t_2} N^*_{j2}\lambda^0_j\,, \qquad \tilde{H}^0_d = \sum_{t_2} N^*_{j3}\lambda^0_j\,, \qquad \tilde{H}^0_u = \sum_{t_2} N^*_{j4}\lambda^0_j \tag{F.13}$$

b) Mass matrix for charginos, basis: $\left(\tilde{W}^-, \tilde{H}^-_d\right) / \left(\tilde{W}^+, \tilde{H}^+_u\right)$

$$m_{\tilde{\chi}^-} = \begin{pmatrix} M_2 & \frac{1}{\sqrt{2}} g_2 v_u \\ \frac{1}{\sqrt{2}} g_2 v_d & \mu \end{pmatrix} \tag{F.14}$$

This matrix is diagonalized by U and V

$$U^* m_{\tilde{\chi}^-} V^\dagger = m_{\tilde{\chi}^-}^{dia} \tag{F.15}$$

with

$$\tilde{W}^- = \sum_{t_2} U^*_{j1}\lambda^-_j\,, \qquad \tilde{H}^-_d = \sum_{t_2} U^*_{j2}\lambda^-_j\,, \qquad \tilde{W}^+ = \sum_{t_2} V^*_{1j}\lambda^+_j\,, \qquad \tilde{H}^+_u = \sum_{t_2} V^*_{2j}\lambda^+_j \tag{F.16}$$

c) Mass matrix for leptons, basis: $(e_{L,o_1}) / \left(e^*_{R,p_1}\right)$

$$m_e = \left(\frac{1}{\sqrt{2}} v_d Y_{e,p_1 o_1} \right) \tag{F.17}$$

This matrix is diagonalized by U^e_L and U^e_R

$$U^{e,*}_L m_e U^{e,\dagger}_R = m_e^{dia} \tag{F.18}$$

with

$$e_{L,i} = \sum_{t_2} U^{e,*}_{L,ji} E_{L,j}\,, \qquad e_{R,i} = \sum_{t_2} U^e_{R,ij} E^*_{R,j} \tag{F.19}$$

d) Mass matrix for down-quarks, basis: $(d_{L,o_1\alpha_1}) / \left(d^*_{R,p_1\beta_1}\right)$

$$m_d = \left(\frac{1}{\sqrt{2}} v_d \delta_{\alpha_1\beta_1} Y_{d,p_1 o_1} \right) \tag{F.20}$$

This matrix is diagonalized by U^d_L and U^d_R

$$U^{d,*}_L m_d U^{d,\dagger}_R = m_d^{dia} \tag{F.21}$$

with

$$d_{L,i\alpha} = \sum_{t_2} U^{d,*}_{L,ji} D_{L,j\alpha}\,, \qquad d_{R,i\alpha} = \sum_{t_2} U^d_{R,ij} D^*_{R,j\alpha} \tag{F.22}$$

e) Mass matrix for up-quarks, basis: $(u_{L,o_1\alpha_1}) \,/\, \left(u^*_{R,p_1\beta_1}\right)$

$$m_u = \left(\begin{array}{c} \frac{1}{\sqrt{2}} v_u \delta_{\alpha_1\beta_1} Y_{u,p_1o_1} \end{array} \right) \tag{F.23}$$

This matrix is diagonalized by U^u_L and U^u_R

$$U^{u,*}_L m_u U^{u,\dagger}_R = m^{dia}_u \tag{F.24}$$

with

$$u_{L,i\alpha} = \sum_{t_2} U^{u,*}_{L,ji} U_{L,j\alpha}\,, \qquad u_{R,i\alpha} = \sum_{t_2} U^u_{R,ij} U^*_{R,j\alpha} \tag{F.25}$$

f) Mass matrix for down-squarks, basis: $\left(\tilde{d}_{L,o_1\alpha_1}/\tilde{d}_{R,o_2\alpha_2}\right), \left(\tilde{d}^*_{L,p_1\beta_1}, \tilde{d}^*_{R,p_2\beta_2}\right)$

$$m_{11} = \frac{1}{24}\delta_{\alpha_1\beta_1}\Big(12\Big(2m^2_{q,o_1p_1} + v_d^2 \sum_{a=1}^{3} Y^*_{d,ap_1} Y_{d,ao_1}\Big) - \Big(3g_2^2 + g_1^2\Big)\Big(-v_u^2 + v_d^2\Big)\delta_{o_1p_1}\Big) \tag{F.26}$$

$$m_{12} = \frac{1}{\sqrt{2}}\delta_{\alpha_1\beta_2}\Big(v_d T_{d,p_2o_1} - v_u \mu^* Y_{d,p_2o_1}\Big) \tag{F.27}$$

$$m_{22} = \frac{1}{12}\delta_{\alpha_2\beta_2}\Big(6\Big(2m^2_{d,p_2o_2} + v_d^2 \sum_{a=1}^{3} Y^*_{d,o_2a} Y_{d,p_2a}\Big) + g_1^2\Big(-v_d^2 + v_u^2\Big)\delta_{o_2p_2}\Big) \tag{F.28}$$

This matrix is diagonalized by Z^D:

$$Z^D m^2_{\tilde{d}} Z^{D,\dagger} = m^{dia}_{2,\tilde{d}} \tag{F.29}$$

with

$$\tilde{d}_{L,i\alpha} = \sum_{t_2} Z^{D,*}_{ji} \tilde{d}_{j\alpha}\,, \qquad \tilde{d}_{R,i\alpha} = \sum_{t_2} Z^{D,*}_{ji} \tilde{d}_{j\alpha} \tag{F.30}$$

g) Mass matrix for sneutrinos, basis: $(\tilde{\nu}_{L,o_1}) \,/\, \left(\tilde{\nu}^*_{L,p_1}\right)$

$$m^2_{\tilde{\nu}} = \left(\begin{array}{c} \frac{1}{8}\Big(8m^2_{l,o_1p_1} + \Big(g_1^2 + g_2^2\Big)\Big(-v_u^2 + v_d^2\Big)\delta_{o_1p_1}\Big) \end{array} \right) \tag{F.31}$$

This matrix is diagonalized by Z^V:

$$Z^V m^2_{\tilde{\nu}} Z^{V,\dagger} = m^{dia}_{2,\tilde{\nu}} \tag{F.32}$$

with

$$\tilde{\nu}_{L,i} = \sum_{t_2} Z^{V,*}_{ji} \tilde{\nu}_j \tag{F.33}$$

h) Mass matrix for up-squarks, basis: $(\tilde{u}_{L,o_1\alpha_1}, \tilde{u}_{R,o_2\alpha_2}) \,/\, \left(\tilde{u}^*_{L,p_1\beta_1}, \tilde{u}^*_{R,p_2\beta_2}\right)$

$$m_{11} = \frac{1}{24}\delta_{\alpha_1\beta_1}\Big(12\Big(2m^2_{q,o_1p_1} + v_u^2\sum_{a=1}^{3} Y^*_{u,ap_1}Y_{u,ao_1}\Big) - \Big(-3g_2^2 + g_1^2\Big)\Big(-v_u^2 + v_d^2\Big)\delta_{o_1p_1}\Big) \tag{F.34}$$

$$m_{12} = \frac{1}{\sqrt{2}}\delta_{\alpha_1\beta_2}\Big(-v_d\mu^* Y_{u,p_2o_1} + v_u T_{u,p_2o_1}\Big) \tag{F.35}$$

$$m_{22} = \frac{1}{6}\delta_{\alpha_2\beta_2}\Big(3v_u^2\sum_{a=1}^{3} Y^*_{u,o_2a}Y_{u,p_2a} + 6m^2_{u,p_2o_2} + g_1^2\Big(-v_u^2 + v_d^2\Big)\delta_{o_2p_2}\Big) \tag{F.36}$$

This matrix is diagonalized by Z^U:

$$Z^U m^2_{\tilde{u}} Z^{U,\dagger} = m^{dia}_{2,\tilde{u}} \tag{F.37}$$

with

$$\tilde{u}_{L,i\alpha} = \sum_{t_2} Z^{U,*}_{ji}\tilde{u}_{j\alpha}\,, \qquad \tilde{u}_{R,i\alpha} = \sum_{t_2} Z^{U,*}_{ji}\tilde{u}_{j\alpha} \tag{F.38}$$

i) Mass matrix for sleptons, basis: $(\tilde{e}_{L,o_1}, \tilde{e}_{R,o_2}) \,/\, \left(\tilde{e}^*_{L,p_1}, \tilde{e}^*_{R,p_2}\right)$

$$m_{11} = \frac{1}{8}\Big(4v_d^2\sum_{a=1}^{3} Y^*_{e,ap_1}Y_{e,ao_1} + 8m^2_{l,o_1p_1} + \Big(-g_2^2 + g_1^2\Big)\Big(-v_u^2 + v_d^2\Big)\delta_{o_1p_1}\Big) \tag{F.39}$$

$$m_{12} = \frac{1}{\sqrt{2}}\Big(v_d T_{e,p_2o_1} - v_u\mu^* Y_{e,p_2o_1}\Big) \tag{F.40}$$

$$m_{22} = \frac{1}{4}\Big(2v_d^2\sum_{a=1}^{3} Y^*_{e,o_2a}Y_{e,p_2a} + 4m^2_{e,p_2o_2} + g_1^2\Big(-v_d^2 + v_u^2\Big)\delta_{o_2p_2}\Big) \tag{F.41}$$

This matrix is diagonalized by Z^E:

$$Z^E m^2_{\tilde{e}} Z^{E,\dagger} = m^{dia}_{2,\tilde{e}} \tag{F.42}$$

with

$$\tilde{e}_{L,i} = \sum_{t_2} Z^{E,*}_{ji}\tilde{e}_j\,, \qquad \tilde{e}_{R,i} = \sum_{t_2} Z^{E,*}_{ji}\tilde{e}_j \tag{F.43}$$

j) Mass matrix for scalar Higgs, basis: (ϕ_d, ϕ_u)

$$m_h^2 = \begin{pmatrix} m^2_{H_d} + |\mu|^2 + \frac{1}{8}\Big(g_1^2 + g_2^2\Big)\Big(3v_d^2 - v_u^2\Big) & -\Re\Big(B_\mu\Big) - \frac{1}{4}\Big(g_1^2 + g_2^2\Big)v_d v_u \\ -\Re\Big(B_\mu\Big) - \frac{1}{4}\Big(g_1^2 + g_2^2\Big)v_d v_u & m^2_{H_u} + |\mu|^2 - \frac{1}{8}\Big(g_1^2 + g_2^2\Big)\Big(-3v_u^2 + v_d^2\Big) \end{pmatrix} \tag{F.44}$$

This matrix is diagonalized by Z^H:

$$Z^H m_h^2 Z^{H,\dagger} = m^{dia}_{2,h} \tag{F.45}$$

with

$$\phi_d = \sum_{t_2} Z^H_{j1} h_j \,, \qquad \phi_u = \sum_{t_2} Z^H_{j2} h_j \tag{F.46}$$

The mixing matrix can be parametrized by

$$Z^H = \begin{pmatrix} -\sin\alpha & \cos\alpha \\ \cos\alpha & \sin\alpha \end{pmatrix} \tag{F.47}$$

k) Mass matrix for pseudo scalar Higgs, basis: (σ_d, σ_u)

$$m^2_{A^0} = \begin{pmatrix} m^2_{H_d} + |\mu|^2 + \frac{1}{8}\Big(g_1^2 + g_2^2\Big)\Big(-v_u^2 + v_d^2\Big) & \Re\Big(B_\mu\Big) \\ \Re\Big(B_\mu\Big) & m^2_{H_u} + |\mu|^2 - \frac{1}{8}\Big(g_1^2 + g_2^2\Big)\Big(-v_u^2 + v_d^2\Big) \end{pmatrix} \tag{F.48}$$

This matrix is diagonalized by Z^A:

$$Z^A m^2_{A^0} Z^{A,\dagger} = m^{dia}_{2,A^0} \tag{F.49}$$

with

$$\sigma_d = \sum_{t_2} Z^A_{j1} A^0_j \,, \qquad \sigma_u = \sum_{t_2} Z^A_{j2} A^0_j \tag{F.50}$$

The mixing matrix can be parametrized by

$$Z^A = \begin{pmatrix} -\cos\beta & \sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \tag{F.51}$$

l) Mass matrix for charged Higgs, basis: $\big(H_d^-, H_u^{+,*}\big)$

$$m^2_{H^-} = \begin{pmatrix} m^2_{H_d} + |\mu|^2 + \frac{1}{8}\Big(g_1^2 + g_2^2\Big)\Big(v_d^2 - v_u^2\Big) & \frac{1}{4} g_2^2 v_d v_u + B_\mu \\ \frac{1}{4} g_2^2 v_d v_u + B_\mu^* & m^2_{H_u} + |\mu|^2 + \frac{1}{8}\Big(g_2^2 - g_1^2\Big)\Big(v_d^2 - v_u^2\Big) \end{pmatrix} \tag{F.52}$$

This matrix is diagonalized by Z^+:

$$Z^+ m^2_{H^-} Z^{+,\dagger} = m^{dia}_{2,H^-} \tag{F.53}$$

with

$$H_d^- = \sum_{t_2} Z^{+,*}_{j1} H_j^- \,, \qquad H_u^+ = \sum_{t_2} Z^+_{j2} H_j^+ \tag{F.54}$$

The mixing matrix can be parametrized by

$$Z^+ = \begin{pmatrix} -\cos\beta & \sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \tag{F.55}$$

F.7 Tadpole equations

$$\frac{\partial V}{\partial v_d} = \frac{1}{8}\Big(8v_d|\mu|^2 - 8v_u\Re\Big(B_\mu\Big) + v_d\Big(8m_{H_d}^2 + g_1^2v_d^2 - g_1^2v_u^2 + g_2^2v_d^2 - g_2^2v_u^2\Big)\Big) \tag{F.56}$$

$$\frac{\partial V}{\partial v_u} = \frac{1}{8}\Big(-8v_d\Re\Big(B_\mu\Big) + 8v_u|\mu|^2 + v_u\Big(8m_{H_u}^2 - g_1^2v_d^2 + g_1^2v_u^2 - g_2^2v_d^2 + g_2^2v_u^2\Big)\Big) \tag{F.57}$$

Appendix G

Verification of Output

We have tested the model files generated by SARAH for the MSSM after electroweak symmetry with the existing model files of FeynArts and CalcHep/CompHep in several ways.

G.1 FeynArts

G.1.1 Comparison on Vertex Level

It is quite easy to translate the standard MSSM model file included in FeynArts to the parametrization used by SARAH by because both are packages for Mathematica. So our first check was comparing the vertices by doing the following steps:

a) Generate a model file from SARAH

b) Read the model file and the standard FeynArts model file

c) Expand all indices (leads to more than 15000 interaction!)

d) Replace the parameters from FeynArts with the corresponding parameters of SARAH

e) Insert numerical values (of SPS 1)

f) Check that the same interaction have the same numerical values at least up to eight digits.

G.1.2 Comparing on Amplitude Squared Level

The second step of checking our FeynArts model file was to calculate the amplitude squared of several processes to make sure that there is no problem with interferences. Here we have checked about 50 processes (more have been done with CalcHep) with different generic structure and compared again the numerical values. For $2 \to 2$ processes we have set $s = (10000 GeV)^2$ and $t = (111 GeV)^2$ to get a numerical values for the amplitude squared.

G.2 CalcHep/MicrOmegas

With CalcHep and MicrOmegas we did the three steps to verify our produced model file. But first, we must say some words about the used model file of CalcHep.

G.2.1 Used Model File for Checks

In the MSSM file used by the current version of `CalcHep` the one loop corrections for the Higgs self interactions are added. Of course, this can't be done by `SARAH`. Furthermore, the interactions of the Goldstone bosons are related to the interaction of the corresponding gauge boson and aren't the interaction of the pseudo scalar.
In order to be comparable with the model file produced by `SARAH` we have used a older version of a MSSM file of `CompHep`, which can be found under `theory.sinp.msu.ru/~semenov/mssm.html`. In this file it is possible to switch the loop corrections off and the Goldstone interaction are the interaction of the first pseudo scalar Higgs.

G.2.2 Comparison on Vertex Level

For comparing the model files on vertex level, we first formatted them in a way to be readable in Mathematica by using a SED script. After that we used relations between the parameters of `CalcHep` and the parameters of `SARAH` and introduces numerical values for the free parameters. One must be aware of the fact that the convention for the mixing matrices in `SARAH` and `CalcHep` differs by a transposition.

G.2.3 Comparison on Process Level

The second step in the validation of our `CalcHep` model was to calculate more than 200 cross sections and decay widths using the Vegas routine of `CalcHep`. About 40 calculations have been made in unitarity- as well as in Feynman-gauge. In addition, more than 20 four point interactions generated by auxiliary fields in `CalcHep` and `SARAH` have been separately checked by switching off the other diagrams.

G.2.4 Comparison of Relic Density

The last step was calculating the relic density with MicrOmegas for SPS 1 and SPS 2. Besides the values for Ωh^2 the contribution of the different annihilation channels was checked: The maximal difference for all channels contributing more than 0.01 % was between the dominating channel of SPS 2 ($\chi_1\chi_1 \to hh$): 30.00% to 29.99%.

G.3 Renormalization Group Equations

We have compared the analytical results for the one and two loop RGEs calculated by `SARAH` for the MSSM with [20], for the NMSSM with [33] and for the MSSM with bilinear R-Parity violation with [34]. In addition, we have checked a model containing non-fundamental representations: the $SU(5)$ inspired Seesaw II model of [35] and [36]. It is know that the there are discrepancies of the RGEs given in this two papers. Our result fully agrees with [35].
Also numerically checks have been done by comparing the RGEs for the MSSM with the RGEs implemented in SPheno [37]. Both set of RGEs were in full agreement.

Appendix H

Evaluation Time

To give a impression for the needed evaluation time for different routines and models, we collected some values in Table H.1. This times were measured under Mathematica 5.2 running on a Intel Quad CPU Q8200 with 2.33 GHz and 4GB RAM.

Command	MSSM (No FV)	MSSM-CKM	NMSSM [1]	$\mu\nu$SSM	MSSM + $U(1)$
`Start`	12.75	18.03	19.02	27.06	16.14
`ModelOutput[EWSB]`	74.83	78.70	94.64	115.08	110.47
`MakeFeynArts`	0.74	3.58	1.12	0.98	0.48
`MakeCalcHep[]`	6.03	22.74	15.57	47.08	6.7
`MakeTeX[]`	0.81	5.79	1.25	1.38	1.48
`CalcRGEs[]`	50.72	50.8	91.07	265.29	68.18
`CalcLoopCorrections[EWSB]`	7.07	28.44	8.14	7.98	8.84
`Vertex[{bar[Fd],Fd,VP}]`	0.05	0.06	0.06	0.06	0.05
`Vertex[{Sd,Sd*,Su,Su*}]`	0.54	0.61	0.56	0.56	0.52
`Vertex[{VG,VG,VG,VG}]`	0.03	0.03	0.04	0.03	0.03
`Vertex[{Se,Se,VZ}]`	0.37	0.37	0.45	1.13	0.37
`Vertex[{bar[gWmC],gP,VWm*}]`	0.02	0.02	0.02	0.02	0.02
`Vertex[{Hpm,VWm*,VZ}]`	0.16	0.17	0.26	0.23	0.21
`TreeMass[VZ,EWSB]`	0.15	0.19	0.20	0.25	0.48
`TreeMass[Su,EWSB]`	0.25	0.27	0.29	0.34	0.23

Table H.1: Time needed in seconds to evaluate several commands of `SARAH` in Mathematica 5.2

[1] With two rotations in the pseudo scalar sector

Bibliography

[1] P. Ramond. Dual theory for free fermions. *Physical Review D*, 3(10):2415–2418, May 1971.

[2] J. Wess and B. Zumino. Supergauge transformations in four-dimensions. *Nuclear Physics*, B70:39–50, 1974.

[3] D. V. Volkov and V. P. Akulov. Is the neutrino a goldstone particle? *Physical Letters*, B46:109–110, 1973.

[4] M. H. Friedman and Y. Srivastava. Allowed supersymmetric vacuums. *Physical Review D*, 16(2):304–306, Jul 1977.

[5] S. Ferrara, J. Wess, and B. Zumino. Supergauge multiplets and superfields. *Physical Letters*, B51:239, 1974.

[6] L. Girardello and Marcus T. Grisaru. Soft breaking of supersymmetry. *Nuclear Physics*, B194:65, 1982.

[7] Steven Weinberg. Implications of dynamical symmetry breaking. *Physical Review D*, 13(4):974–996, Feb 1976.

[8] S. Weinberg. Implications of dynamical symmetry breaking: An addendum. *Physical Review D*, 19(4):1277–1280, Feb 1979.

[9] Paul Langacker and Mingxing Luo. Implications of precision electroweak experiments for m_t, ρ_0, $sin^2\theta_W$, and grand unification. *Physical Review D*, 44(3):817–822, Aug 1991.

[10] John R. Ellis, S. Kelley, and Dimitri V. Nanopoulos. Probing the desert using gauge coupling unification. *Physical Letters*, B260:131–137, 1991.

[11] C. Giunti, C. W. Kim, and U. W. Lee. Running coupling constants and grand unification models. *Mod. Physical Letters*, A6:1745–1755, 1991.

[12] L. Hall. Grand unification of effective gauge theories. *Nuclear Physics B*, 178:75–124, February 1981.

[13] John R. Ellis, J. S. Hagelin, Dimitri V. Nanopoulos, Keith A. Olive, and M. Srednicki. Supersymmetric relics from the big bang. *Nucl. Phys.*, B238:453–476, 1984.

[14] T. Hahn. Generating feynman diagrams and amplitudes with feynarts 3. *Computer Physics Communications*, 140:418, 2001, hep-ph/0012260.

[15] A. Pukhov. CalcHEP 3.2: MSSM, structure functions, event generation, batchs, and generation of matrix elements for other packages. 2004, hep-ph/0412191.

[16] A. S. Belyaev et al. CompHEP-PYTHIA interface: Integrated package for the collision events generation based on exact matrix elements. 2000, hep-ph/0101232.

[17] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov. Dark matter direct detection rate in a generic model with micrOMEGAs2.1. *Comput. Phys. Commun.*, 180:747–767, 2009, 0803.2360.

[18] Werner Porod. Spheno, a program for calculating supersymmetric spectra, susy particle decays and susy particle production at e+ e- colliders. *Computer Physics Communications*, 153:275, 2003, hep-ph/0301101.

[19] B. Allanach et al. SUSY Les Houches Accord 2. *Comp. Phys. Commun.*, 180:8–25, 2009, 0801.0045.

[20] Stephen P. Martin and Michael T. Vaughn. Two loop renormalization group equations for soft supersymmetry breaking couplings. *Phys. Rev.*, D50:2282, 1994, hep-ph/9311340.

[21] Damien M. Pierce, Jonathan A. Bagger, Konstantin T. Matchev, and Ren-jie Zhang. Precision corrections in the minimal supersymmetric standard model. *Nucl. Phys.*, B491:3–67, 1997, hep-ph/9606211.

[22] T. Hahn and M. Perez-Victoria. Automatized one-loop calculations in 4 and d dimensions. *Computer Physics Communications*, 118:153, 1999, hep-ph/9807565.

[23] G. Belanger, Neil D. Christensen, A. Pukhov, and A. Semenov. SLHAplus: a library for implementing extensions of the standard model. 2010, 1008.0181.

[24] Edward Witten. An SU(2) anomaly. *Phys. Lett.*, B117:324–328, 1982.

[25] Youichi Yamada. Two loop renormalization of gaugino masses in general supersymmetric gauge models. *Phys. Rev. Lett.*, 72:25–27, 1994, hep-ph/9308304.

[26] I. Jack, D. R. T. Jones, and A. Pickering. The soft scalar mass beta-function. *Phys. Lett.*, B432:114–119, 1998, hep-ph/9803405.

[27] Ian Jack, D. R. Timothy Jones, Stephen P. Martin, Michael T. Vaughn, and Youichi Yamada. Decoupling of the epsilon scalar mass in softly broken supersymmetry. *Phys. Rev.*, D50:5481–5483, 1994, hep-ph/9407291.

[28] D. R. T. Jones and L. Mezincescu. The Chiral Anomaly and a Class of Two Loop Finite Supersymmetric Gauge Theories. *Phys. Lett.*, B138:293, 1984.

[29] Peter C. West. The Yukawa beta Function in N=1 Rigid Supersymmetric Theories. *Phys. Lett.*, B137:371, 1984.

[30] I. Jack, D. R. T. Jones, and A. Pickering. Renormalisation invariance and the soft beta functions. *Phys. Lett.*, B426:73–77, 1998, hep-ph/9712542.

[31] Youichi Yamada. Two loop renormalization of gaugino mass in supersymmetric gauge model. *Phys. Lett.*, B316:109–111, 1993, hep-ph/9307217.

[32] Youichi Yamada. Two loop renormalization group equations for soft SUSY breaking scalar interactions: Supergraph method. *Phys. Rev.*, D50:3537–3545, 1994, hep-ph/9401241.

[33] Ulrich Ellwanger, Cyril Hugonie, and Ana M. Teixeira. The Next-to-Minimal Supersymmetric Standard Model. 2009, 0910.1785.

[34] B. C. Allanach, A. Dedes, and H. K. Dreiner. The R parity violating minimal supergravity model. *Phys. Rev.*, D69:115002, 2004, hep-ph/0309196.

[35] Francesca Borzumati and Toshifumi Yamashita. Minimal supersymmetric SU(5) model with non-renormalizable operators: Seesaw mechanism and violation of flavour and CP. 2009, 0903.2793.

[36] Anna Rossi. Supersymmetric seesaw without singlet neutrinos: Neutrino masses and lepton-flavour violation. *Phys. Rev.*, D66:075003, 2002, hep-ph/0207006.

[37] Werner Porod. SPheno, a program for calculating supersymmetric spectra, SUSY particle decays and SUSY particle production at e+ e- colliders. *Comput. Phys. Commun.*, 153:275–315, 2003, hep-ph/0301101.