PROCEEDINGS OF SCIENCE

Tutorial to SARAH

Florian Staub[*]
Theoretical Physics Department, CERN, Geneva, Switzerland
E-mail: florian.staub@cern.ch

I give in this tutorial a short practical introduction to the Mathematica package SARAH. First, it is shown how an existing model file can be changed to implement a new model in SARAH. In the second part, masses, vertices and renormalisation group equation are calculated with SARAH.

Proceedings of the Corfu Summer Institute 2015 "School and Workshops on Elementary Particle Physics and Gravity"
1-27 September 2015
Corfu, Greece

[*]Speaker.

1. Introduction

The Mathematica package SARAH [2, 3, 4, 5, 6] was created to simplify the study of new models: SARAH has been optimized for an easy, fast and exhaustive study of non-minimal SUSY and non-SUSY models. It can calculate all important tree-level properties of the model, i.e. mass matrices and vertices, and interfacing this information to Monte-Carlo (MC) tools, but calculates also one-loop self-energies and two-loop renormalization group equations (RGEs). Using this information, SARAH became the first 'spectrum-generator-generator': all analytical information derived by SARAH can be exported to Fortran code which provides a fully-fledged spectrum generator based on SPheno [7, 8]. A SPheno version created in that way calculates a large number of flavour observables using the FlavorKit [9], and provides even Higgs masses at the two-loop level for a given model [10, 11]. An detailed overview to these calculations has been given in the lecture at this school [12], and many more details about the framework are given in Ref. [1]. I focus now on a short practical introduction to SARAH.
First, I briefly demonstrate how an existing model in SARAH can quickly be changed to a new model, and how the model file is used for basic calculations. I assume in the following that SARAH is located in the directory [$SARAH].
The simple example which is considered in the following is to extent the superpotential of the NMSSM to include dimensionful terms

$$W = \cdots + \mu \hat{H}_d \hat{H}_u + \frac{1}{2} \mu_s \hat{S} \hat{S} \tag{1.1}$$

Afterwards, I show how the new model can be used and how information about the model is calculated with SARAH.

2. Changing the model file

To get started we copy the model directory of the NMSSM to a new subdirectory called CorfuNMSSM and rename the file NMSSM.m in this directory to CorfuNMSSM.m:

```
> cp -R [\$SARAH]/Models/NMSSM    [\$SARAH]/Models/CorfuNMSSM
> mv [\$SARAH]/Models/CorfuNMSSM/NMSSM.m
                    [\$SARAH]/Models/CorfuNMSSM/CorfuNMSSM.m
```

The next step is to add the new terms to the superpotential. For this purpose, we open CorfuNMSSM.m in a text editor and change the line which defines the superpotential to

```
SuperPotential = Yu u.q.Hu - Yd d.q.Hd - Ye e.l.Hd + \
           \[Lambda] Hu.Hd.s + \[Kappa]/3 s.s.s \
                 + Mu Hu.Hd + 1/2 MuS s.s;
```

In order to have a better looking LaTeX output in the following, we slightly extend parameters.m also included in the directory CorfuNMSSM:

```
ParameterDefinitions = {
```

```
{Mu, {LaTeX -> "\\mu"}},
{B[Mu], {LaTeX -> "B_\\mu"}},
{MuS, {LaTeX -> "\\mu_S"}},
{B[MuS], {LaTeX -> "B_S"}},

{g1,        { Description -> "Hypercharge-Coupling"}},
...
```

We are already done and we can run the model now.

3. Running SARAH

We open Mathematica and run the new model via

```
<<[\$SARAH]/SARAH.m;
Start["CorfuNMSSM"];
```

We find the expected warning from SARAH that the define Z_3 symmetry is not respected in the new terms:

```
Superpotential::ViolationGlobal: Warning!
   Global symmetry Z3 not conserved in {Hu,Hd}
Superpotential::ViolationGlobal: Warning!
   Global symmetry Z3 not conserved in {s,s}
```

However, we can ignore this warning and continue with the study. After a few seconds, SARAH is finished with the initialization of the model.

4. Calculating analytical properties

4.1 Tree-level properties and loop corrections

After the initialization of the model, we can for instance check that the new terms appear in all mass matrices. For this purpose, we run

```
 MassMatrix[Chi]
```

and get the neutralino mass matrix which reads

$$\begin{pmatrix} \mathrm{MassB} & 0 & -\frac{\mathrm{g1vd}}{2} & \frac{\mathrm{g1vu}}{2} & 0 \\ 0 & \mathrm{MassWB} & \frac{\mathrm{g2vd}}{2} & -\frac{\mathrm{g2vu}}{2} & 0 \\ -\frac{\mathrm{g1vd}}{2} & \frac{\mathrm{g2vd}}{2} & 0 & -\mathrm{Mu}-\frac{\lambda \mathrm{vS}}{\sqrt{2}} & -\frac{\lambda \mathrm{vu}}{\sqrt{2}} \\ \frac{\mathrm{g1vu}}{2} & -\frac{\mathrm{g2vu}}{2} & -\mathrm{Mu}-\frac{\lambda \mathrm{vS}}{\sqrt{2}} & 0 & -\frac{\lambda \mathrm{vd}}{\sqrt{2}} \\ 0 & 0 & -\frac{\lambda \mathrm{vu}}{\sqrt{2}} & -\frac{\lambda \mathrm{vd}}{\sqrt{2}} & \mathrm{MuS}+\sqrt{2}\kappa \mathrm{vS} \end{pmatrix} \tag{4.1}$$

In the same way, we can check the Higgs mass matrices MassMatrix[hh] and MassMatrix[Ah] and find that there not only MuS and Mu shows up, but also the automatically generated soft-breaking terms B[MuS] and B[Mu].

The minimisation conditions of the model this model are returned via

```
TadpoleEquations[EWSB]
```

Again, one sees that the news terms are consistently taken into account everywhere.

We can now either calculate specific vertices like for instance the Higgs-down-quark vertex via

```
Vertex[{bar[Fu],Fu,hh}]
```

and get the expected result

$$\Big\{\{\mathrm{bar}(\mathrm{Fu}(\{\mathrm{gt1},\mathrm{ct1}\})),\mathrm{Fu}(\{\mathrm{gt2},\mathrm{ct2}\}),\mathrm{hh}(\{\mathrm{gt3}\})\},\\ \Big\{-\frac{i\mathrm{ZH}(\mathrm{gt3},2)\mathrm{Delta}(\mathrm{ct1},\mathrm{ct2})\mathrm{sum}(\mathrm{j2},1,3,\mathrm{conj}(\mathrm{ZUL}(\mathrm{gt2},\mathrm{j2}))\mathrm{sum}(\mathrm{j1},1,3,\mathrm{Yu}(\mathrm{j1},\mathrm{j2})\mathrm{conj}(\mathrm{ZUR}(\mathrm{gt1},\mathrm{j1}))))}{\sqrt{2}},\mathrm{PL}\Big\},\\ \Big\{-\frac{i\mathrm{ZH}(\mathrm{gt3},2)\mathrm{Delta}(\mathrm{ct1},\mathrm{ct2})\mathrm{sum}(\mathrm{j2},1,3,\mathrm{ZUL}(\mathrm{gt1},\mathrm{j2})\mathrm{sum}(\mathrm{j1},1,3,\mathrm{ZUR}(\mathrm{gt2},\mathrm{j1})\mathrm{conj}(\mathrm{Yu}(\mathrm{j1},\mathrm{j2}))))}{\sqrt{2}},\mathrm{PR}\Big\}\Big\}$$

or we can calculate all vertices at once. For this purpose we run[1]

```
ModelOutput[EWSB, IncludeLoopCorrections -> True,
 VerticesForLoops -> True,WriteTeX -> True]
```

Here, we calculate not only the vertices for the eigenstates after electroweak symmetry breaking (EWSB), but we also (i) include the vertices needed for the loop calculations, (ii) calculate the one-loop corrections, and (iii) generate LaTeX files with all information. After 1–2 minutes SARAH is finished. We can now check for instance all cubic and quartic vertices which are saved in the arrays

```
SA`VertexList[SSS]
SA`VertexList[SSSS]
```

However, this is a bit lengthy and difficult to read. Therefore, we check the LaTeX output. For this purpose we enter the output directory and make the shell script to run pdflatex executable

```
cd [\$SARAH]\Output/CorfuNMSSM/EWSB/TeX/
chmod 775 MakePDF.sh
./MakePDF.sh
```

This script does not only run pdflatex but generates also all Feynman diagrams. After some seconds we get a pdf file with 122 pages which contains all information about the model which SARAH has derived so far.

[1] Some warnings will appear here and in the following because of missing definitions for μ, μ_S, B_S and B_μ. The reason is that, in order to save time, we did not put all information for these new parameters in parameters.m which are possible. However, these warnings can be ignored for many outputs.

4.2 RGEs

The next step is to check the RGEs for this model. To do this, we just have to run

```
CalcRGEs[];
```

After not even one minute, SARAH is finished with the calculation of all two-loop β-functions. We can check the RGEs for the new superpotential terms via

```
BetaMuij
```

and get

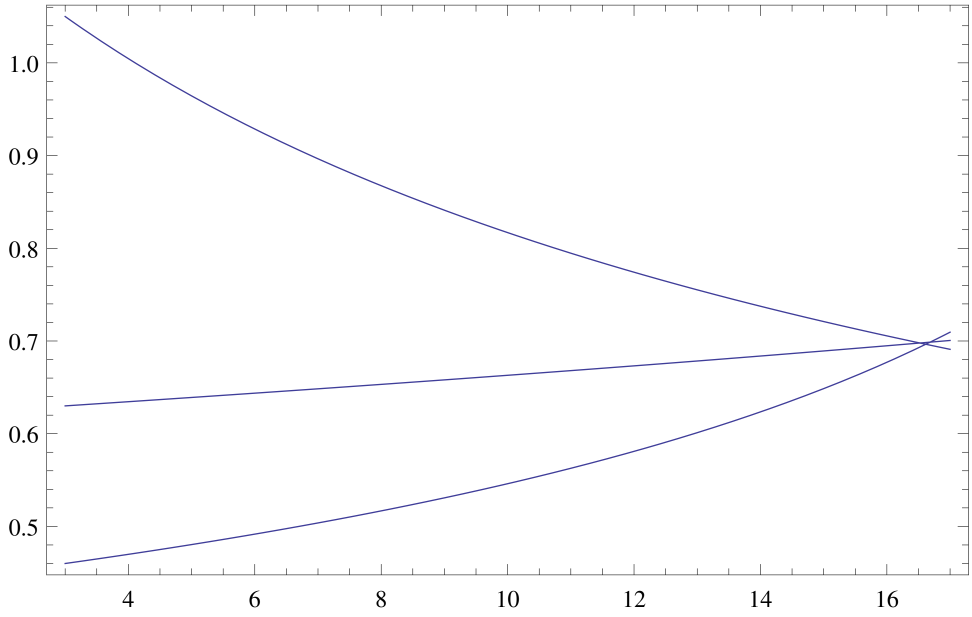
```
{{Mu,
  -((3 g1^2 Mu)/5) - 3 g2^2 Mu + 2 Mu \[Lambda] conj[\[Lambda]] +
     3 Mu trace[Yd, Adj[Yd]] + Mu trace[Ye, Adj[Ye]] +
     3 Mu trace[Yu, Adj[Yu]],
  1/50 Mu (207 g1^4 + 90 g1^2 g2^2 + 375 g2^4 -
     200 \[Kappa] \[Lambda] conj[\[Kappa]] conj[\[Lambda]] -
     300 \[Lambda]^2 conj[\[Lambda]]^2 - 20 g1^2 trace[Yd, Adj[Yd]] +
     800 g3^2 trace[Yd, Adj[Yd]] + 60 g1^2 trace[Ye, Adj[Ye]] +
     40 g1^2 trace[Yu, Adj[Yu]] + 800 g3^2 trace[Yu, Adj[Yu]] -
     50 \[Lambda] conj[\[Lambda]] (3 trace[Yd, Adj[Yd]] +
        trace[Ye, Adj[Ye]] + 3 trace[Yu, Adj[Yu]]) -
     450 trace[Yd, Adj[Yd], Yd, Adj[Yd]] -
     300 trace[Yd, Adj[Yu], Yu, Adj[Yd]] -
     150 trace[Ye, Adj[Ye], Ye, Adj[Ye]] -
     450 trace[Yu, Adj[Yu], Yu, Adj[Yu]])},
{MuS,
   4 MuS (\[Kappa] conj[\[Kappa]] + \[Lambda] conj[\[Lambda]]),
 -(4/5) MuS (20 \[Kappa]^2 conj[\[Kappa]]^2 +
     20 \[Kappa] \[Lambda] conj[\[Kappa]] conj[\[Lambda]] + \[Lambda] \
     conj[\[Lambda]] (-3 g1^2 - 15 g2^2 + 10 \[Lambda] conj[\[Lambda]] +
     15 trace[Yd, Adj[Yd]] + 5 trace[Ye, Adj[Ye]] +
     15 trace[Yu, Adj[Yu]]))}}
```

For both parameter the one- and two-loop β-function is given up to factors $1/16\pi^2$.

SARAH writes also a file to run these RGEs numerically with Mathematica. One can load this file in any Mathematica session and use the provided function RunRGEs to solve the RGEs numerically. We are going to use this to check if we can reproduce gauge coupling unification:

```
<< [\$SARAH]/Output/CorfuNMSSM/RGEs/RunRGEs.m;
sol = RunRGEs[{g1 -> 0.46, g2 -> 0.63, g3 -> 1.05}, 3, 17][[1]];
Plot[{g1[x], g2[x], g3[x]} /. sol, {x, 3, 17}, Frame -> True, Axes -> False];
```

We first loaded the file and then run the RGEs from 10^3 to 10^{17} GeV. As initial conditions at 1 TeV we used $g_1 = 0.46$, $g_2 = 0.63$, $g_3 = 1.05$. The interpolation function were saved in the variable `sol` which we then used to make a plot. The result is not very surprising:



5. Outputs for other codes

Since already all vertices are calculated, we can start now to generate model files for the different tools. The corresponding commands are

- `MakeCHep[]` for `CalcHep/CompHep`
- `MakeFeynArts[]` for `FeynArts/FormCalc`
- `MakeUFO[]` for UFO
- `MakeWHIZARD[]` for `WHIZARD/O'Mega`
- `MakeVevacious[]` for `Vevacious`

Before we can use

```
MakeSPheno[]
```

to generate the Fortran code to implement this model in `SPheno`, we would have slightly to modify `[$SARAH]/Models/CorfuNMSSM/SPheno.m` to define the boundary conditions for the new parameters. However, this is beyond the scope of this short tutorial and I refer interested user to Ref. [1] for detailed discussions and examples how to use these commands.

6. Summary

I have shown, how a new SUSY singlet extenions can be implemented easily `SARAH` starting from the default MSSM model file. I then described how analytical and numerical results can quickly be obtained for this model.

Acknowledgements

I thank the organizers of “Summer School and Workshop on the Standard Model and Beyond”, Corfu2015, for the invitation and the hospitality during the stay.

References

[1] F. Staub, (2015), 1503.04200.

[2] F. Staub, (2008), 0806.0538.

[3] F. Staub, Comput.Phys.Commun. **181** (2010), 1077–1086, [0909.2863].

[4] F. Staub, Comput.Phys.Commun. **182** (2011), 808–833, [1002.0840].

[5] F. Staub, Computer Physics Communications **184** (2013), pp. 1792–1809, [1207.0906].

[6] F. Staub, (2013), 1309.7223.

[7] W. Porod, Comput. Phys. Commun. **153** (2003) 275 doi:10.1016/S0010-4655(03)00222-4 [hep-ph/0301101].

[8] W. Porod and F. Staub, Comput. Phys. Commun. **183**, 2458 (2012) doi:10.1016/j.cpc.2012.05.021 [arXiv:1104.1573 [hep-ph]].

[9] W. Porod, F. Staub, and A. Vicente, Eur.Phys.J. **C74** (2014), 2992, [1405.1434].

[10] M. D. Goodsell, K. Nickel, and F. Staub, Eur. Phys. J. **C75** (2015), no. 1, 32, [1411.0675].

[11] M. Goodsell, K. Nickel, and F. Staub, Eur. Phys. J. **C75** (2015), no. 6, 290, [1503.03098].

[12] F. Staub, arXiv:1509.07061 [hep-ph].