

SARAH in a nutshell

October 28, 2015

For more details, explanations and examples, check

Exploring new models in all detail with SARAH, arXiv:1503.04200

and the manual, of course.

What is SARAH?

SARAH is a *Mathematica* package for building and analysing SUSY and non-SUSY models. It calculates all vertices, mass matrices, tadpoles equations, one-loop corrections for tadpoles and self-energies, and two-loop RGEs for a given model. SARAH writes model files for *FeynArts*, *CalcHep/CompHep*, which can also be used for dark matter studies using *MicrOmegas*, in the *UFO* format which is supported by *MadGraph*, *Herwig++* and *Sherpa*, as well as for *WHIZARD* and *O'Mega*.

SARAH is also the first available spectrum-generator-generator: based on the derived, analytical expression it creates source code for *SPheno*. In that way, it is possible to implement new models in *SPheno* without the need to write any Fortran code by hand. The output for *Vevacious* can be used to check for the global minimum for a given model and parameter point. Running SARAH is considerably fast, it includes already a long list of SUSY and non-SUSY models, and the implementation of new models is efficient and straightforward.

How do I install SARAH?

The installation of SARAH is very simple: the package can be downloaded from

<http://sarah.hepforge.org>

After copying the `tar` file to the directory `$PATH`, it can be extracted

```
$ cp [Download-Directory]/SARAH-X.Y.Z.tar.gz $PATH/  
$ cd $PATH  
$ tar -xf SARAH-X.Y.Z.tar.gz  
$ ln -s SARAH SARAH-X.Y.Z
```

`X.Y.Z` has to be replaced by the version which was downloaded. In the last line a symbolic link from the directory `SARAH` to `SARAH-X.Y.Z` is created. There is no compilation necessary, but SARAH can directly be used with any *Mathematica* version between 7 and 10.

How do I load a model in SARAH?

To load an existing model (called `$MODEL` in the following) run in *Mathematica*

```
<<[$PATH]/SARAH.m;  
Start["$MODEL"];
```

After about 1 minute the message

```
| All Done. $MODEL is ready!
```

should appear and no error messages or warning during the evaluation should show up.

How do I get a spectrum generator based on SPheno for a new model?

To obtain the SPheno output run in Mathematica after SARAH is loaded and the model is initialized the command

```
| MakeSPheno [] ;
```

When executing MakeSPheno, SARAH calculates first all information which it needs, i.e. it is not necessary that the user has done the calculation of vertices or RGEs before. When SARAH is done, the source code for SPheno is stored in \$SPATH/SARAH/Output/\$MODEL/EWSB/SPheno/. The compilation of this codes is done as follows: one has to enter the directory of the SPheno installation, a new sub-directory has to be created and the code must be copied into this directory:

```
$ cd $PATH/SPHENO
$ mkdir $MODEL
$ cp $PATH/SARAH/Output/$MODEL/EWSB/SPheno/* $MODEL/
```

Afterwards, the code is compiled via

```
$ make Model=$MODEL
```

and a new executable SPheno\$MODEL is available.

An input file, by default called LesHouches.in.\$MODEL, is needed to run SPheno\$MODEL. SARAH writes a template for that file which has been copied to the \$MODEL subdirectory of SPheno together with the Fortran code. You can move it to the root directory of SPheno

```
$ cp \ $MODEL/LesHouches.in.\ $MODEL .
```

By doing this we can work now from the SPheno main directory and we don't have to give the file as argument when running SPheno. Thus, SPheno can be called via

```
$ ./bin/SPheno\ $MODEL
```

Alternatively, one can keep the Les Houches file in the \$MODEL and work with it via

```
$ ./bin/SPheno\ $MODEL \ $MODEL/LesHouches.in.\ $MODEL
```

Of course, before you can run SPheno you must first fill the template written by SARAH with numbers.

What are the features of a SPheno version generated like this?

A spectrum generator based on SARAH/SPheno provides the same precision as dedicated MSSM codes do for a huge variety of models. The NMSSM version provides even a more precise Higgs mass calculation than other public NMSSM codes do. The main features are

- Full two-loop running of all parameters and all masses at one-loop
- Complete one-loop thresholds at M_Z from BSM to gauge and Yukawa couplings
- Two-loop corrections to Higgs masses
- calculation of flavour and precision observables at full one-loop
- calculation of decay widths and branching ratios
- interface to HiggsBounds and HiggsSignals
- estimate of electroweak Fine-Tuning

How can I check points with HiggsBounds and HiggsSignals?

In the same directory in which the SPheno spectrum file is located, also all other input files for HiggsBounds and HiggsSignals are saved by SPheno. The (relative) path to this directory has to be given as last argument to HiggsBounds when executing it. Thus, working from the directory \$PATH, HiggsBounds is started via:

```
$ ./HiggsBounds/HiggsBounds LandH effC #neutral #charged SPHENO/
```

From other directories, one can use absolute paths:

```
$ $PATH/HiggsBounds/HiggsBounds LandH effC #neutral #charged ↔  
$PATH/SPHENO/
```

#neutral and *#charged* are the number of (physical) neutral and charged Higgs bosons in the model. **HiggsSignals** is the complement to **HiggsBounds** and checks how good a point reproduces the Higgs mass and rate measurements. The syntax is very similar to **HiggsBounds**:

```
$ ./HiggsSignals/HiggsSignals latestresults peak 2 effC #neutral ↔  
#charged SPHENO/
```

How do I implement a model in MadGraph and link it to SPheno

A new model in **MadGraph** can be implemented via the UFO format which is also supported by other tools like **Herwig++** or **Sherpa**. The command to generate the UFO files is

```
| MakeUFO []
```

All files written by **SARAH** have to be copied to a new sub-directory in **MadGraph**'s model directory:

```
cd $PATH/MADGRAPH/  
mkdir models/$MODEL  
cp $PATH/SARAH/Output/$MODEL/EWSB/UFO/* models/$MODEL
```

The model can be loaded in **MadGraph** via

```
import model $MODEL --modelname
```

The option `--modelname` is used to keep the names of the particles as given in the model files. This prevents conflicts with internal **MadGraph** conventions.

How do I implement a model in WHIZARD/O'Mega and link it to SPheno

The model files for **WHIZARD/O'Mega** are obtained by

```
| MakeWHIZARD []
```

After the interface has completed, the generated files are stored in the directory

```
$PATH/SARAH/Output/$MODEL/EWSB/WHIZARD_Omega/
```

In order to use the model with **WHIZARD** and **O'Mega**, the generated code must be compiled and installed. In most cases this is done by

```
$ cd $PATH/SARAH/Output/$MODEL/EWSB/WHIZARD.Omega  
$ ./configure --prefix=$PATH/WHIZARD/ WO_CONFIG=$PATH/WHIZARD/bin/  
$ make  
$ make install
```

If **WHIZARD** has not been installed globally in the home directory of the current user, **WHIZARD** won't be able to find the binaries. Thus, the `WO_CONFIG` environment variable was used to point explicitly to the binaries. By default, the `configure` script would install the compiled model into `.whizard` in the home directory of the user. If the user wants to have several **WHIZARD** installations or install **WHIZARD** locally, it might be better to provide a model just for one installation. For these cases the installation path has been defined via the `--prefix` option of the `configure` script. More information on the available options is shown with the command

```
./configure --help
```

To link WHIZARD and SPheno, all SPheno modules created by SARAH write the information about the parameters and masses into an additional file. This file is written in the WHIZARD specific format and can be directly read by WHIZARD. One just has to make sure that the corresponding flag is turned on the Les Houches input for SPheno to get this output:

LesHouches.in.\$MODEL

```
Block SPhenoInput      # SPheno specific input
...
75 1                    # Write WHIZARD files
```

The parameter file can then be included in the Sindarin input file for WHIZARD via

```
model = $model_sarah
...
include("$PATH/SPHENO/WHIZARD.par.$MODEL")
```

How can I calculate the relic density with MicrOmegas

The calculation of the cross section and all necessary decay widths are done by CalcHep which comes together with MicrOmegas. Thus, a new model in MicrOmegas is implemented by providing the corresponding CalcHep model files. That means, one can use the SARAH output for CalcHep to work with MicrOmegas:

```
| MakeCHep [] ;
```

When SARAH is finished with MakeCHep, the CalcHep model files are located in the directory

```
$PATH/SARAH/Output/$MODEL/EWSB/CHep/
```

To implement the model in MicrOmegas, a new project has to be created and the files have to be copied in the working directory of this project:

```
$ cd $PATH/MICROMEAS
$ ./newProject MODEL
$ cd $MODEL
$ cp $PATH/SARAH/Output/$MODEL/EWSB/CHep/* work/models
```

Steering files written by SARAH were copied together with all model files into the working directory of the current project. You can move them to the main project directory and compile it

```
$ mv work/models/CalcOmega_with_DDetection.cpp .
$ make main=CalcOmega_with_DDetection.cpp
```

A new binary CalcOmega_with_DDetection is now available. The only missing piece are the input parameters.

Providing the numerical parameters is pretty easy because MicrOmegas/CalcHep can read the SPheno spectrum file. However, the user must make sure that no complex rotation matrices show up in the spectrum file: in the case of Majorana matrices and no CP violation, there are two equivalent outputs: (i) all Majorana masses are positive, but some entries of the corresponding rotation matrices are complex; (ii) all mixing matrices are real, but some masses are negative. CalcHep can just handle the second case with real matrices. Hence, one has to use the flag

LesHouches.in.BLSSM

```
Block SPhenoInput      # SPheno specific input
...
50 0                    # Majorana phases: use only positive masses
```

to get the spectrum according to that convention. Afterwards, the spectrum file just has to be moved to the same directory as CalcOmega_with_DDetection. Copy it there and start the calculation:

```
$ cp $PATH/SPHENO/SPheno.spc.$MODEL .
$ ./CalcOmega_with_DDetection
```

How do I implement new model in Vevacious?

The model files for `Vevacious` are generated via

```
| MakeVevacious [] ;
```

As soon as the model file is created, it is convenient to copy them to the model directory of the local `Vevacious` installation. In addition, one can also generate a new subdirectory which contains the `SPheno` spectrum files for the `$MODEL` used as input for `Vevacious`, as well as the output written by `Vevacious`

```
$ cd $PATH/VEVACIOUS
$ mkdir $MODEL/
$ cp $PATH/SARAH/Output/$MODEL/Vevacious/$MODEL.vin models/
$ cp $PATH/SPHENO/SPheno.spc.$MODEL $MODEL/
```

These steps are just optional: the user can give in the initialization file used by `Vevacious`, which is discussed in a second, also paths to other locations of the model and spectrum file. Independent of the location of the files, one has to write this initialization file for a new study. The easiest way is to start with the file included in the `Vevacious` package in the subdirectory `bin` and edit it

```
$ cd $PATH/VEVACIOUS/bin
$ cp VevaciousInitialization.xml VevaciousInitialization_.$MODEL.xml
```

How do I get the model files for FeynArts/FormCalc?

Run

```
MakeFeynArts []
```

How do I get all expression in L^AT_EX format?

Use

```
| MakeTeX [] ;
```

When `SARAH` is done with the output, the `.tex` file are stored in
`$PATH/SARAH/Output/$MODEL/EWSB/TeX`

The main file which can be compiled with `pdflatex` is `$MODEL.EWSB.tex`. In order to compile all Feynman diagrams `SARAH` provides also a shell script `MakePDF.sh`.