Tools for HEP

From $\mathcal{L}$ to Observables

Alexander Voigt
alexander.voigt@physik.rwth-aachen.de

RWTH Aachen

Workshop on
"Selected topics on future directions in particle physics"
Warsaw 2024

Table of Contents

Material

Material for the workshop:

https://github.com/Expander/material-heptools-2024-warsaw

Slides for the workshop:

https://users.hepforge.org/~avoigt/download/Warschau-2024.pdf

Overview

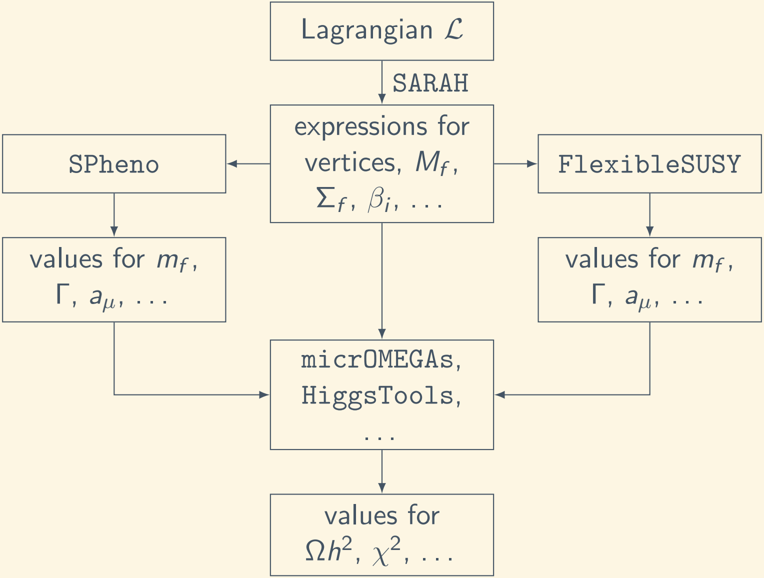

Lagrangian $\mathcal{L}$
SARAH
expressions for vertices, M_f, Σ_f, β_i, ...
SPheno
FlexibleSUSY
values for m_f, Γ, a_μ, ...
values for m_f, Γ, a_μ, ...
micrOMEGAs, HiggsTools, ...
values for Ωh^2, χ^2, ...

Table of Contents

Standard Model – What we will do

1. Inspect the `SARAH` model files and generate some analytic expressions
2. Create a `SPheno` spectrum generator
3. Create a `FlexibleSUSY` spectrum generator

Table of Contents

Standard Model – SARAH – inspect model files

Go to the `SARAH` directory:

```
$ cd ~/hep-software/SARAH
```

Let's inspect the following `SARAH` files:

- `Models/SM/SM.m`
  Definitions of the gauge group, fields, Lagrangian, VEVs, field mixings, ...
- `Models/SM/particles.m`
  Definitions of properties of the fields (output name, LaTeX name, PDG number, FeynArts number, ...)
- `Models/SM/parameters.m`
  Definitions of information about the model parameters (Output name, LaTeX name, SLHA output block, ...)

References: `https://sarah.hepforge.org/`

Standard Model – SARAH – inspect model properties

Let's start SARAH and inspect some model properties.

Start Mathematica:

```
$ math
```

Run

```
In[]:= << SARAH`
In[]:= << Start["SM"]
In[]:= SARAH`Commands (* shows available commands *)
```

Let's inspect some model properties:

```
In[]:= SARAHparameters (* parameters *)
In[]:= Particles[EWSB] (* fields after EWSB *)
In[]:= MassMatrix[Fe]  (* mass matrix of Fe *)
In[]:= Vertex[{bar[Fe],Fe,VP}] (* a vertex *)
```

Standard Model – SARAH – generate some expressions

Let's generate some analytic expressions.

```
In[]:= MakeAll[] (* generate all expressions *)
In[]:= Quit[] (* quit *)
```

The expressions are written to the `Output/SM/` sub-directory.

Vertices (fermion–fermion–gauge boson):

```
$ less Output/SM/EWSB/Vertices/VertexListFFV.m
```

Beta functions (gauge couplings):

```
$ less Output/SM/RGEs/BetaGauge.m
```

1-loop self-energies:

```
$ less Output/SM/EWSB/One-Loop/SelfEnergy.m
```

Table of Contents

Standard Model – SPheno – spectrum generator

Let's create a SPheno spectrum generator for the SM.

Let's first look into the SPheno model file for the SM:

```
$ less Models/SM/SPheno.m
```

It defines boundary conditions for input and output parameters, loop corrections, observables, ...

Now we'll create the FORTRAN source code for the SPheno spectrum generator:

```
$ math
In[]:= << SARAH`
In[]:= Start["SM"]
In[]:= MakeSPheno[]
In[]:= Quit[]
```

Standard Model – SPheno – spectrum generator

Now go to the SPheno directory and copy the generated FORTRAN files to the SM sub-directory.

```
$ cd ~/hep-software/SPheno-4.0.5
$ mv ~/hep-software/SARAH/Output/SM/EWSB/SPheno SM
```

Then we compile the source code:

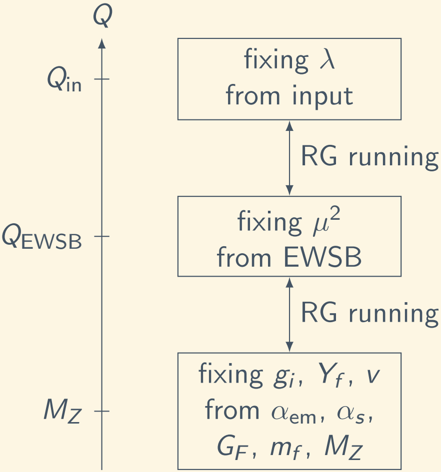
```
$ make F90=gfortran
$ make F90=gfortran Model=SM
```

Now we can run the spectrum generator and inspect the SLHA output file:

```
$ bin/SPhenoSM SM/Input_Files/LesHouches.in.SM
$ less SPheno.spc.SM
```

Table of Contents

Standard Model – FlexibleSUSY – spectrum generator



Standard Model – FlexibleSUSY – spectrum generator

Let's now create a `FlexibleSUSY` spectrum generator for the SM.

Go into the FlexibleSUSY-2.8.0 directory:

```
$ cd ~/hep-software/FlexibleSUSY-2.8.0
```

Let's first look into the `FlexibleSUSY` model file for the SM:

```
$ less model_files/SM/FlexibleSUSY.m.in
```

It defines boundary conditions for input and output parameters, loop corrections, observables, ...

References:
https://arxiv.org/abs/1406.2319
https://arxiv.org/abs/1710.03760

Standard Model – FlexibleSUSY – spectrum generator

Now we'll build the `FlexibleSUSY` spectrum generator for the SM.

Define a variable to contain the `LoopTools` directory

```
$ LT=$HOME/hep-software/LoopTools/build
```

Create and build the SM spectrum generator:

```
$ ./createmodel --name=SM
$ ./configure --with-models=SESM \
      --with-loop-libraries=looptools \
      --with-looptools-incdir=${LT}
      --with-looptools-libdir=${LT}
$ make -j4
```

Run the spectrum generator (set `FlexibleSUSY[31] = 2`):

```
$ models/SM/run_SM.x \
    --slha-input-file=models/SM/LesHouches.in.SM
```

Table of Contents

Standard Model + S

As a second example, let's consider an extension of the SM by an extra real scalar gauge singlet. The model should be called Singlet Extension of the Standard Model (SESM). The Lagrangian should read

$$\mathcal{L}_{\mathrm{SESM}} = \mathcal{L}_{\mathrm{SM}} - \left[\kappa_{SH} H^\dagger H s + \frac{\lambda_{SH}}{2} H^\dagger H s^2 + \frac{M_S^2}{2} s^2 + \frac{\kappa_S}{3} s^3 + \frac{\lambda_S}{2} s^4\right]$$

We will proceed as follows:

1. Create a `SARAH` model file for the SESM
2. Create a `FlexibleSUSY` model file for the SESM and run it
3. Create a `SPheno` model file for the SESM and run it

Table of Contents

Standard Model + S – SARAH – model file

First, we will create a SARAH model file for our new SESM model.

Go into the SARAH directory:

```
$ cd ~/hep-software/SARAH
```

We start by copying SARAH's SM model files to a new directory named SESM:

```
$ mkdir -p Models/SESM
$ cp Models/SM/parameters.m Models/SESM/
$ cp Models/SM/particles.m Models/SESM/
$ cp Models/SM/SM.m Models/SESM/SESM.m
```

Standard Model + S – SARAH – model file

First, we modify `Models/SESM/SESM.m`:

1. Add the new scalar field:

```
ScalarFields[[2]] = {s, 1, Sing, 0, 1, 1};
RealScalars = {s};
```

3. Extend the Lagrangian:

```
LagNoHC = -(
    + mu2 conj[H].H
    + \[Lambda]/2 conj[H].H.conj[H].H
    + KapSH conj[H].H.s
    + LamSH/2 conj[H].H.s.s
    + MS2/2 s.s
    + KapS/3 s.s.s
    + LamS/2 s.s.s.s
);
```

Standard Model + S – SARAH – model file

4. Give a VEV to the new (real) singlet:

$$s = v_S + \phi_S$$

```
DEFINITION[EWSB][VEVs] = {
  {H0, {v, 1/Sqrt[2]},
       {Ah, \[ImaginaryI]/Sqrt[2]},
       {phiH, 1/Sqrt[2]} },
  {Sing, {vS, 1},
         {0, 0},
         {phiS, 1}
};
```

Standard Model + S – SARAH – model file

5. Mix ϕ_S with ϕ_H:

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = Z_H \begin{pmatrix} \phi_H \\ \phi_S \end{pmatrix}, \qquad Z_H^T = Z_H$$

```
DEFINITION[EWSB][MatterSector] = {
  {{phiH, phiS}, {hh, ZH}},
  {{{dL}, {conj[dR]}}, {{DL,Vd}, {DR,Ud}}},
  {{{uL}, {conj[uR]}}, {{UL,Vu}, {UR,Uu}}},
  {{{eL}, {conj[eR]}}, {{EL,Ve}, {ER,Ue}}}
};
```

Standard Model + S – SARAH – model file

6. Modify the particle information in `Models/SESM/particles.m`.

First, define the new `Sing` gauge eigenstate

```
ParticleDefinitions[GaugeES] = {
  {Sing, { Description -> "Singlet",
           PDG -> {0},
           Width -> 0,
           Mass -> Automatic,
           ElectricCharge -> 0,
           FeynArtsNr -> 3,
           LaTeX -> "s",
           OutputName -> "s" }},
  ...
};
```

Standard Model + S – SARAH – model file

6. . . . continuing . . .

Extend Higgs to be a doublet in mass eigenstates

```
ParticleDefinitions[EWSB] = {
  {hh, { Description -> "Higgs",
         PDG -> {25,35}, (* <-- two PDG numbers *)
         PDG.IX -> {101000001,101000002} (* <-- *)
       }
  },
  ...
};
```

Standard Model + S – SARAH – model file

6. ... continuing ...

Change the name of the intermediate neutral component of the SM Higgs doublet to phiH:

```
WeylFermionAndIndermediate = {
  {phiH, { PDG -> {0}, (* <-- name changed to phiH *)
           Width -> 0,
           Mass -> Automatic,
           LaTeX -> "H",
           OutputName -> "" }},
  ...
};
```

Standard Model + S – SARAH – model file

7. Modify the parameter information in `Models/SESM/parameters.m`.

Add the new model parameters and their properties

```
ParameterDefinitions = {
  {ZH, { Description -> "Scalar-Mixing-Matrix" }},
  {\[Alpha],
       { Description -> "Scalar mixing angle" }},
  {vS, { Dependence -> None,
         DependenceNum -> None,
         DependenceOptional -> None,
         DependenceSPheno -> None,
         Real -> True,
         LesHouches -> {HMIX, 51},
         LaTeX -> "vS",
         OutputName -> vS }},
  ...
};
```

Standard Model + S – SARAH – model file

7. . . . continuing . . .

```
ParameterDefinitions = {
  ...
  {MS2, { LaTeX -> "M_S^2",
          Real -> True,
          OutputName -> MS2,
          LesHouches -> {HMIX,30} }},
  {LamS, { LaTeX -> "\\lambda_S",
           Real -> True,
           OutputName -> LamS,
           LesHouches -> {HMIX,31} }},
  {LamSH, { LaTeX -> "\\lambda_{SH}",
            Real -> True,
            OutputName -> LamSH,
            LesHouches -> {HMIX,32} }},
  ...
};
```

Standard Model + S – SARAH – model file

7. . . . continuing . . .

```
ParameterDefinitions = {
  ...
  {KapS, { LaTeX -> "\\kappa_S",
           Real -> True,
           OutputName -> KapS,
           LesHouches -> {HMIX,33} }},
  {KapSH, { LaTeX -> "\\kappa_{SH}",
            Real -> True,
            OutputName -> KapSH,
            LesHouches -> {HMIX,34} }}
};
```

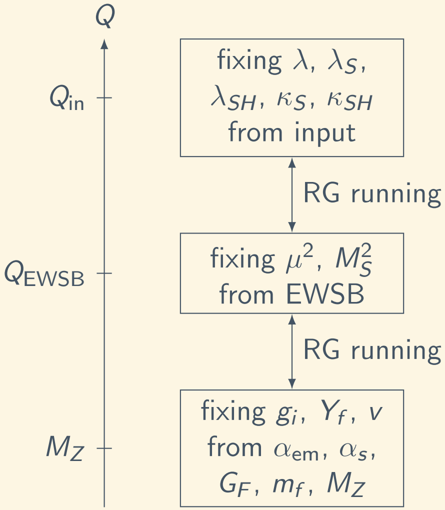
Standard Model + S – SARAH – check model

8. Check the SARAH model

```
$ math
In[]:= << SARAH`
In[]:= Start["SESM"]
In[]:= CheckModel[]
In[]:= Quit[]'
```

Table of Contents

Standard Model + S – FlexibleSUSY – spectrum generator



Standard Model + S – FlexibleSUSY – model file

Now we turn to create a `FlexibleSUSY` model file.

1. Go to the `FlexibleSUSY` directory:

```
$ cd ~/hep-software/FlexibleSUSY-2.8.0
```

Copy the SM model to a new model directory named `model_files/SESM`

```
$ mkdir -p model_files/SESM
$ cp model_files/SM/FlexibleSUSY.m.in \
    model_files/SESM/
$ cp model_files/SM/LesHouches.in.SM \
     model_files/SESM/LesHouches.in.SESM
```

Standard Model + S – FlexibleSUSY – model file

2. Edit the `FlexibleSUSY` model file `model_files/SESM/FlexibleSUSY.m.in`

Define new input parameters and the SLHA blocks they are read from:

```
FSDefaultSARAHModel = SESM; (* SARAH model *)

MINPAR = {};

EXTPAR = {
    {0, Qin},
    {1, QEWSB},
    {2, LambdaIN},
    {3, LamSIN},
    {4, LamSHIN},
    {5, KapSIN},
    {6, KapSHIN},
    {7, vSIN}
};
```

Standard Model + S – FlexibleSUSY – model file

2. ... continuing ...

Fix the new parameters at appropriate scales by EWSB and by the input values:

```
EWSBOutputParameters = { mu2, MS2 };

HighScaleInput = {
    {\[Lambda], LambdaIN},
    {LamS, LamSIN},
    {LamSH, LamSHIN},
    {KapS, KapSIN},
    {KapSH, KapSHIN}
};

SUSYScaleInput = {
    {vS, vSIN}
};
```

Delete the SM-specific loop corrections.

Standard Model + S – FlexibleSUSY – build

3. Create, configure and compile the FlexibleSUSY spectrum generator:

```
$ LT=$HOME/hep-software/LoopTools/build

$ ./createmodel -f --name=SESM

$ ./configure --with-models=SESM \
      --with-loop-libraries=looptools \
      --with-looptools-incdir=${LT} \
      --with-looptools-libdir=${LT}

$ make -j4
```

Standard Model + S – FlexibleSUSY – SLHA input

4. Set reasonable values for the input parameters in the SLHA input file `model_files/SESM/LesHouches.in.SESM`

```
Block EXTPAR         # Input parameters
    0   1000         # input scale Qin
    1   173.34       # scale QEWSB
    2   0.21         # LambdaIN
    3   0.1          # LamSIN
    4   0.1          # LamSHIN
    5   100          # KapSIN
    6  -100          # KapSHIN
    7   3            # vSIN
```

Standard Model + S – FlexibleSUSY – run

5. Run FlexibleSUSY:

```
$ models/SESM/run_SESM.x \
    --slha-input-file=model_files/SESM/LesHouches.in.\
   SESM
```

Table of Contents

Standard Model + S – SPheno – model file

Finally, let's create a SPheno spectrum generator.

1. Go to the SARAH directory and copy the model file SPheno.m from the SM to the SESM model directory:

```
$ cd ~/hep-software/SARAH
$ cp Models/SM/SPheno.m Models/SESM/
```

Standard Model + S – SPheno – model file

2. Modify the `SPheno.m` model file to define new input parameters:

```
MINPAR = {
    {2, LambdaIN},
    {3, LambdaSIN},
    {4, LambdaSHIN},
    {5, KappaSIN},
    {6, KappaSHIN},
    {7, vevSIN}
};
```

Standard Model + S – SPheno – model file

2. … continuing …

Fix the new parameters by EWSB and the input values :

```
ParametersToSolveTadpoles = {mu2, MS2};

BoundaryLowScaleInput = {
    {\[Lambda], LambdaIN},
    {LambdaS, LambdaSIN},
    {LambdaSH, LambdaSHIN},
    {KapS, KappaSIN},
    {KapSH, KappaSHIN},
    {vS, vevSIN}
};
```

Standard Model + S – SPheno – model file

3. Create the FORTRAN files for the SPheno spectrum generator:

```
$ math
In[]:= << SARAH`
In[]:= Start["SESM"]
In[]:= MakeSPheno[]
In[]:= Quit[]'
```

Standard Model + S – SPheno – model file

4. Go to the SPheno directory:

```
$ cd ~/hep-software/SPheno-4.0.5
```

Copy the generated FORTRAN files from the SARAH directory to the SPheno directory and compile the spectrum generator:

```
$ mv ~/hep-software/SARAH/Output/SESM/EWSB/SPheno \
    SESM
$ make F90=gfortran
$ make F90=gfortran Model=SESM
```

Standard Model + S – SPheno – model file

5. Set the input parameters to reasonable values in the SLHA input file `SESM/Input_Files/LesHouches.in.SESM`:

```
Block MINPAR    # Input parameters
 2   0.21       # LambdaIN
 3   0.1        # LamSIN
 4   0.1        # LamSHIN
 5   100        # KapSIN
 6  -100        # KapSHIN
 7   3          # vSIN
```

6. Run the `SPheno` spectrum generator:

```
$ ./bin/SPhenoSSM \
     SESM/Input_Files/LesHouches.in.SESM
```

and inspect the output:

```
$ less SPheno.spc.SESM
```

Table of Contents

Standard Model + S + S

As a second example, let us consider an extension of the SESM by another real scalar gauge singlet. The model should be called Two Singlet Extension of the Standard Model (TSESM). The Lagrangian should have a global Z_2 symmetry with all non-SESM fields having Z_2 charge -1:

$$\mathcal{L}_{\text{TSESM}} = \mathcal{L}_{\text{SESM}} - \left[\frac{\tilde{\lambda}_{SH}}{2} H^\dagger H \tilde{s}^2 + \frac{\tilde{M}_S^2}{2} \tilde{s}^2 + \frac{\tilde{\lambda}_S}{2} \tilde{s}^4\right]$$

We will proceed as follows:

1. Create a `SARAH` model file for the TSESM
2. Create a `FlexibleSUSY` model file for the TSESM
3. Create a `micrOMEGAs` model file for the TSESM
4. Run `FlexibleSUSY` and pass the output to `micrOMEGAs` to calculate the dark matter relic abundance Ωh^2 etc.
5. Pass the SLHA output of `micrOMEGAs` to SModelS.

Table of Contents

Standard Model + S + S – SARAH – model file

First, we modify `Models/TSESM/TSESM.m`:

1. Add a global Z_2 symmetry and assign the charge $+1$ to each SESM field:

```
Global[[1]] = {Z[2], Z2};

Gauge[[1]]={..., 1};
Gauge[[2]]={..., 1};
Gauge[[3]]={..., 1};

FermionFields[[1]] = {..., 1};
FermionFields[[2]] = {..., 1};
FermionFields[[3]] = {..., 1};
FermionFields[[4]] = {..., 1};
FermionFields[[5]] = {..., 1};

ScalarFields[[1]] = {..., 1};
ScalarFields[[2]] = {..., 1};
```

Standard Model + S + S – SARAH – model file

2. Add the new scalar field $\tilde{s}$ with Z_2 charge -1:

```
ScalarFields[[3]] = {ts, 1, TSing, 0, 1, 1, -1};
RealScalars = {s, ts};
```

3. Extend the Lagrangian:

```
LagNoHC = -(
    ...
    + TLamSH/2 conj[H].H.ts.ts
    + TMS2/2 ts.ts
    + TLamS/2 ts.ts.ts.ts
);
```

Standard Model + S + S – SARAH – model file

4. Modify the particle information in `Models/TSESM/particles.m`.

First, define the new `TSing` EWSB eigenstate (= gauge eigenstate)

```
ParticleDefinitions[EWSB] = {
  ...
  {TSing, { Description -> "Singlet",
            PDG -> {6666635},
            PDG.IX -> {101000002},
            FeynArtsNr -> 10,
            Mass -> LesHouches,
            LaTeX -> "\\tilde{s}",
            ElectricCharge -> 0,
            LHPC -> {"gold"},
            OutputName -> "ts" }}
};
```

Standard Model + S + S – SARAH – model file

5. Modify the parameter information in `Models/TSESM/parameters.m`.

Add the new model parameters and their properties

```
ParameterDefinitions = {
  ...
  {TMS2, { Description -> "Singlet mass term",
           LaTeX -> "\\tilde{M}_{S}^2",
           Real -> True,
           OutputName -> TMS2,
           LesHouches -> {HMIX,35} }},

  {TLamSH, { OutputName -> TLamSH,
             LaTeX -> "\\tilde{\\lambda}_{SH}",
             Real -> True,
             LesHouches -> {HMIX,36} }},
  ...
};
```

Standard Model + S + S – SARAH – model file

5. … continuing …

```
ParameterDefinitions = {
  ...
  {TLamS, { OutputName -> TLamS,
            LaTeX -> "\\tilde{\\lambda}_{S}",
            Real -> True,
            LesHouches -> {HMIX,37} }}
};
```

Standard Model + S + S – SARAH – check model

6. Check the SARAH model

```
$ math
In[]:= << SARAH`
In[]:= Start["TSESM"]
In[]:= CheckModel[]
In[]:= Quit[]'
```

Table of Contents

Standard Model + S + S – FlexibleSUSY – model file

Now we create a FlexibleSUSY model file for the TSESM.

1. Go to the FlexibleSUSY directory:

```
$ cd ~/hep-software/FlexibleSUSY-2.8.0
```

Copy the SESM model to a new model directory named model_files/TSESM

```
$ mkdir -p model_files/TSESM
$ cp model_files/SESM/FlexibleSUSY.m.in \
    model_files/TSESM/
$ cp model_files/SESM/LesHouches.in.SM \
     model_files/TSESM/LesHouches.in.TSESM
```

Standard Model + S + S – FlexibleSUSY – model file

2. Edit the `FlexibleSUSY` model file `model_files/TSESM/FlexibleSUSY.m.in`

Define new input parameters and the SLHA blocks they are read from and set the new parameters to their input value:

```
FSDefaultSARAHModel = TSESM; (* SARAH model *)

EXTPAR = {
    ...
    {8, TLamSIN},
    {9, TLamSHIN},
    {10, TMS2IN}
};

HighScaleInput = {
    ...
    {TLamS, TLamSIN},
    {TLamSH, TLamSHIN},
    {TMS2, TMS2IN}
};
```

Standard Model + S + S – FlexibleSUSY – build

3. Create, configure and compile the FlexibleSUSY spectrum generator:

```
$ LT=$HOME/hep-software/LoopTools/build

$ ./createmodel -f --name=TSESM

$ ./configure --with-models=TSESM \
      --with-loop-libraries=looptools \
      --with-looptools-incdir=${LT} \
      --with-looptools-libdir=${LT}

$ make -j4
```

Standard Model + S + S – FlexibleSUSY – SLHA input

4. Set reasonable values for the new input parameters in the SLHA input file `model_files/TSESM/LesHouches.in.TSESM`

```
Block EXTPAR        # Input parameters
    ...
    8   0.1         # TLamSIN
    9   0.1         # TLamSHIN
   10   1e3         # TMS2IN
```

Standard Model + S + S – FlexibleSUSY – run

5. Run FlexibleSUSY:

```
$ models/TSESM/run_TSESM.x \
    --slha-input-file=model_files/TSESM/LesHouches.in\
   .TSESM
```

Predicted Higgs masses from the output:

```
Block MASS
   6666635     6.18452698E+01   # TSing
        25     1.23989800E+02   # hh(1)
        35     6.90311044E+02   # hh(2)
```

Table of Contents

Standard Model + S + S – micrOMEGAs

Our plan is now to pass the FlexibleSUSY output to micrOMEGAs.

1. In order to setup micrOMEGAs for the TSESM, we need to first generate appropriate micrOMEGAs/CalcHEP model files with SARAH:

```
$ cd ~/hep-software/SARAH
$ math
In[]:= << SARAH`
In[]:= Start["TSESM"]
In[]:= MakeCHep[DMcandidate1 -> Z2 == -1, \
    IncludeEffectiveHiggsVertices -> False]
In[]:= Quit[]
```

Standard Model + S + S – micrOMEGAs

2. We go to the `micrOMEGAs` directory and setup the new TSESM model:

```
$ cd ~/hep-software/micromegas_6.1.15
$ ./newProject TSESM
$ cd TSESM
$ cp ~/hep-software/SARAH/Output/TSESM/EWSB/CHep/* \
    work/models/
$ make main=main.cpp
```

Standard Model + S + S – micrOMEGAs

3. We run FlexibleSUSY and place the SLHA output in the current directory with the name `SPheno.spc.TSESM`

```
$ FS=$HOME/hep-software/FlexibleSUSY-2.8.0

$ ${FS}/models/TSESM/run_TSESM.x \
      --slha-input-file=${FS}/model_files/TSESM/\
    LesHouches.in.TSESM \
      > SPheno.spc.TSESM
```

4. We run `micrOMEGAs`:

```
$ ./main data.par
```

Table of Contents

Standard Model + S + S – SModelS

Our plan is now to pass the `micrOMEGAs` SLHA output to SModelS.

In order to generate an SLHA output from `micrOMEGAs`, we have to add the following line to `~/hep-software/micromegas_6.1.15/TSESM/main.cpp`:

```
smodels(LHC8+LHC13, 5, 0.,
    const_cast<char*>("smodels.slha"),
    const_cast<char*>("3.0.0"), 0);
```

We re-compile `micrOMEGAs` and re-run the point:

```
$ make main=main.cpp
$ ./main data.par
```

The SLHA output has now been written to `smodels.slha`.

Standard Model + S + S – SModelS

W. Kotlarski has prepared a custom program under
https://github.com/Expander/material-heptools-2024-warsaw/blob/main/TSESM/CalcOmega-1DM.cpp
Compile and run it as follows:

```
cd ~/hep-software/micromegas_6.1.15
mkdir nlohmann && cd nlohmann
wget https://github.com/nlohmann/json/releases/\
    download/v3.11.3/json.hpp
cd -
make main=CalcOmega-1DM.cpp
./CalcOmega-1DM
```

The SLHA output has now been written to `smodels.slha`.