3 Lecture 2: Computing dark matter properties with `MicrOmegas`

3.1 What is `MicrOmegas`?

`MicrOmegas` is probably the most popular computer tool for the study of dark matter. First developed to compute the relic density of a stable massive particle, the code also computes the rates for direct and indirect detection rates of dark matter. Nowadays, many phenomenologists and dark matter model builders use it on a daily basis.

3.2 `MicrOmegas`: Technical details, installation and load

- **Name of the tool:** `MicrOmegas`
- **Author:** Geneviève Bélanger, Fawzi Boudjema, Alexander Pukhov and Andrei Semenov. They can be contacted via micromegas@lapth.cnrs.fr.
- **Type of code:** `C` and `Fortran`
- **Website:** `https://lapth.cnrs.fr/micromegas/`
- **Manual:** The manual for the latest version of `MicrOmegas` can be found in [17]. For previous versions see [18–21].

After downloading the package, one should copy the `tar.gz` file into the `$PATH` folder and extract its contents,

```
$ cp Download-Directory/micromegas_X.Y.Z.tar.gz $PATH/
$ cd $PATH
$ tar -xf micromegas_X.Y.Z.tar.gz
```

Here `X.Y.Z` must be replaced by the `MicrOmegas` version which has been downloaded. The next step is the compilation of the code, performed with the commands

```
$ cd micromegas_X.Y.Z
$ make
```

And `MicrOmegas` will be ready to run our own dark matter studies.

3.3 General usage and description of the input files

Before we describe the input files generated by `SARAH`, it is convenient to explain the general usage of `MicrOmegas`. This will clarify the role of each file.

Strictly speaking, `MicrOmegas` is not a code, but a collection of routines for the evaluation of dark matter properties. It contains many model-independent functions and routines, which can be used for the specific models we are studying. The way this is done is quite simple. The user must write a short *steering file*, or main program, that (i) defines options for the DM calculations and output, and (ii) calls the built-in routines in `MicrOmegas` that run the desired DM calculations. Hence, the user does not need to enter into the details of the `MicrOmegas` routines, but just call them with the proper options. `MicrOmegas` will then read the details of the input model (contained in external `mdl` files, to be provided for each model) and execute the routines, returning the DM properties (such as relic density and detection rates) required by the user in the steering file.

Let us now describe the input files. As already explained in Sec. 2.5, `SARAH` can produce model files for `MicrOmegas`. Thanks to this feature, the user gets rid of the most tedious task when working with `MicrOmegas`. Once generated, these files are located in the directory `$PATH/SARAH-X.Y.Z/Output/Scotogenic/EWSB/CHep`. One finds the following files:

- `CalcOmega.cpp`
- `CalcOmega_with_DDetection.cpp`
- `CalcOmega_with_DDetection_old.cpp`
- `func1.mdl`

- `lgrng1.mdl`
- `prtcls1.mdl`
- `vars1.mdl`

As explained above, the `mdl` files define the input model, with details such as particle content, interactions and parameters. On the other hand, the `cpp` files are steering files that tell `MicrOmegas` what dark matter properties we are interested in. Although we will not have to get into the details of these files (since `SARAH` did it for us), let us briefly review their content.

The `mdl` files contain information about the model. In the file `vars1.mdl` one finds the definition of several decay widths of the particles in the model (including the SM ones) and standard parameters like the Fermi constant G_F. The file `func1.mdl` is devoted to the *constrained variables* of the model, this is, to all masses and vertices. The Feynman rules are given in the file `lgrng1.mdl`. Notice that each interaction vertex is given in terms of the interacting states, the Lorentz structure and the value of the vertex itself, using for the former the list of vertices defined in `func1.mdl`. For example, we see that the $\eta_I - \eta_I - h$ Feynman rule is given by the `v0002` vertex, which is defined to be equal to $-(\lambda_3 + \lambda_4 - \lambda_5)\, v$ in `func1.mdl`. Also note that the parameters of the model use the names we introduced in the `parameters.m` file using the `OutputName` option. Finally, the file `prtcls1.mdl` contains information about the particles in the model.

The `cpp` files are the main programs, in this case `C` programs, containing all the calculations we want `MicrOmegas` to perform. We can forget about `CalcOmega_with_DDetection_old.cpp`, which is an old version, no longer required. The difference between the other two `cpp` files is whether direction detection rates should be computed or not.

The file `CalcOmega_with_DDetection.cpp` includes the calculation of direct detection rates. We will talk a little about this possibility in Sec. 3.5. However, for our example we will use the file `CalcOmega.cpp`, which only computes the dark matter relic density, $\Omega_{\mathrm{DM}} h^2$.

3.4 Running `MicrOmegas`

In order to implement our model in `MicrOmegas`, we must create a new project and copy the files to the corresponding folder. This is done with

```
$ cd $PATH/micromegas_X.Y.Z
$ ./newProject Scotogenic
$ cd Scotogenic
$ cp $PATH/SARAH-X.Y.Z/Output/Scotogenic/EWSB/CHep/* work/models
```

The next step is the compilation of the selected `cpp` file. However, before we do that, let us notice one thing. In the directory `$PATH/micromegas_X.Y.Z/Scotogenic` one can find two files, `main.c` and `main.F`, with example programs for `MicrOmegas`. They are equivalent to the `cpp` files generated by `SARAH` and contain examples of calculations one can perform with `MicrOmegas`. Therefore, although we will not use them in this course, it might be helpful to take a look at them in order to see the different options and how to turn on specific calculations and outputs in `MicrOmegas`.

In order to compile our own `MicrOmegas` code for the scotogenic model we execute

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

This will create the binary file `CalcOmega` in the `Scotogenic` folder. In order to run it and get our results there is only one thing missing: input parameters. For this purpose we will make use of a very convenient feature of `MicrOmegas`: it can read a spectrum file in LesHouches format. Therefore, we can use `SPheno`, run with the input values of our choice, and pass the resulting output file to `MicrOmegas`, which can then read it and compute the DM observables.

TIP: It has become common nowadays to combine different codes. This makes the study of a model a more efficient task, since using the output of a code as input for another code solves many conversion and formatting issues. For this reason, it is convenient to choose computer tools which can be easily combined.

We already learned how to run `SPheno` in Sec. 2.6. There is only one detail that we must take into account in case we want to use the `SPheno` output file as input for `MicrOmegas`. `MicrOmegas` cannot handle rotation matrices with complex entries. Since these may appear in some calculations with Majorana fermions (like the neutrinos in the scotogenic model), we must tell `SPheno` that we want numerical results without them. Indeed, it is common to find purely imaginary rotation matrices, or rows of them, in models with Majorana fermions even in the absence of CP violating phases. This type of complex phases can be absorbed by adding a negative sign to the mass of the Majorana fermion. This can be easily understood by looking at the transformation between the mass matrix in the gauge basis (M) and the diagonal mass matrix in the mass basis ($\hat{M}$). For a Majorana fermion, this transformation is of the form

$$V\,M\,V^T = \hat{M}\,, \tag{22}$$

where V is a unitary matrix. It is clear that multiplying a row of the V matrix by the imaginary unit i is equivalent to a change of sign in one eigenvalue of $\hat{M}$. Therefore, it is just a matter of convention whether we present the results with complex rotation matrices and positive masses or with real rotation matrices and negative masses. `MicrOmegas` can only understand the input if we take the second option, and thus we must tell `SPheno` to produce an output file with this choice. This is done by setting the flag `50` in the LesHouches input file of `SPheno` to the value `0`, as we already did in Sec. 2.6.

After this comment, we can proceed to run `MicrOmegas` in the benchmark point **BS1** of the scotogenic model. In order to do this, we must copy the `SPheno` spectrum file to the `Scotogenic` folder in `MicrOmegas` and execute the binary file we just created

```
$ cp $PATH/SPheno-X.Y.Z/SPheno.spc.Scotogenic .
$ ./CalcOmega
```

The first time we run the binary it can take some time, even up to several hours depending on the computer power, since `MicrOmegas` has to compile all necessary annihilation channels of the DM candidate for that particular parameter point. All further evaluations of similar points are done in a second or less.

When the run is finished, we get the results on the screen:

```
Masses of odd sector Particles:
~N1  : MN1  =    345.0 || ~etI : MetI  =    430.1 || ~etR : MetR  =    430.1
~etp : Metp =    447.7 || ~N2  : MN2   =   4800.0 || ~N3  : MN3   =   6800.0

Xf=2.41e+01 Omega h^2=1.08e+00

# Channels which contribute to 1/(omega) more than 1%.
# Relative contributions in % are displayed
   28% ~N1 ~N1 ->e3 E3
   21% ~N1 ~N1 ->nu2 nu3
   15% ~N1 ~N1 ->nu2 nu2
    8% ~N1 ~N1 ->e2 E3
    8% ~N1 ~N1 ->E2 e3
    7% ~N1 ~N1 ->nu3 nu3
    4% ~N1 ~N1 ->nu1 nu2
    3% ~N1 ~N1 ->nu1 nu3
    2% ~N1 ~N1 ->e2 E2
```

First, `MicrOmegas` writes the masses (in GeV) of all particles charged under the $\mathbb{Z}_2$ parity. Note that their names are written including a tilde (~), in contrast to the names of the $\mathbb{Z}_2$-even particles, which do not have it. Since the lightest $\mathbb{Z}_2$-odd particle in the **BS1** point is the lightest right-handed neutrino, N_1, it is stable and constitutes the dark matter of the universe.

Next, `MicrOmegas` gives us two quantities. $x_f = m_{N_1}/T_f$ characterizes the freeze-out temperature, T_f, and $\Omega_{\mathrm{DM}}h^2$ is the dark matter relic density. We see that in the benchmark point **BS1** we obtain $\Omega_{\mathrm{DM}}h^2 = 1.08$. This relic density is too high, since the Planck observations prefer a value in the $\Omega_{\mathrm{DM}}h^2 \sim 0.11$ ballpark. Therefore, this parameter point is also excluded due to DM constraints.

`MicrOmegas` also gives a list with the annihilation channels that give the most relevant contributions to the DM relic density. In the **BS1** point, the most important one is

$$N_1 N_1 \to \tau^+\tau^- \tag{23}$$

which constitutes 28% of the total annihilation cross-section. Note also that flavor violating channels are present as well in the list. For example, we find that the channels

$$N_1 N_1 \to \mu^\pm \tau^\mp \tag{24}$$

contribute with 16% of the annihilation cross-section.

Finally, note that this information is exported to the external files `omg.out` and `channels.out`. These two files are written following the LesHouches format: each entry is defined by a flag (or a few of them) and a numerical value.

3.5 Other computations in `MicrOmegas`

In the previous Section we learned how to use the `CalcOmega.cpp` file which is automatically provided by `SARAH`. With the aid of this file we can easily compute the DM relic density. However, it is easy to modify these files to (i) change some details of these calculations, (ii) change the amount of information that is shown as output, and (iii) compute additional observables.

We already saw that we can compute direct detection rates with the file `CalcOmega_with_DDetection.cpp`. This file is the main file for a `C` program that uses `MicrOmegas` to calculate the DM relic density $\Omega_{\mathrm{DM}} h^2$ as well as some direct detection rates: (i) spin independent cross-section with proton and neutron in pb, (ii) spin dependent cross-section with proton and neutron in pb, (iii) recoil events in the 10 - 50 keV region at ^{73}Ge, ^{131}Xe, ^{23}Na and ^{127}I nuclei. We decided not to use this file in benchmark point **BS1** because, for this parameter point, the DM scattering cross-sections with nucleons is zero at tree-level. Therefore, we would have obtained vanishing direct detection rates.

Just to see how direct detection rates are obtained, let us consider a slight modification of the **BS1** benchmark point. The reason why the benchmark point **BS1** leads to vanishing direct detection rates at tree-level is because the DM particle in this point is the lightest right-handed neutrino and this state does not couple directly to the nucleons. Instead, in a parameter point with scalar DM (η_I), the tree-level scattering cross-section with the nucleons does not vanish. Therefore, let us define a new benchmark point, **BS2** (Benchmark Scotogenic 2), with a lighter η_I state. The only change with respect to the **BS1** point is:

$$\boxed{m_\eta^2 = 5 \cdot 10^4 \,\mathrm{GeV}^2}$$

Using this parameter point is straightforward. We just have to modify a single line in the `MINPAR` block of the `LesHouches.in.Scotogenic` input file:

LesHouches.in.Scotogenic

```
6    5.000000E+04     # mEt2Input
```

After running `SPheno` with this modification in the input file, we generate a new `SPheno.spc.Scotogenic` output file that we can use with `MicrOmegas`. We can easily check that this parameter indeed leads to a much lighter η_I state:

SPheno.spc.Scotogenic

```
1002      2.23606798E+02   # etI
```

Now we can create the `CalcOmega_with_DDetection` binary. This is completely analogous to what we did for the `CalcOmega` binary:

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

Once compiled, we copy our new `SPheno.spc.Scotogenic` file and run the `CalcOmega_with_DDetection` binary file,

```
$ cp $PATH/SPheno-X.Y.Z/SPheno.spc.Scotogenic .
$ ./CalcOmega_with_DDetection
```

This is the result that is printed on the screen:

```
Masses of odd sector Particles:
~etI : MetI  =   223.6 || ~etR : MetR  =   223.6 || ~etp : Metp  =   255.8
~N1  : MN1   =   345.0 || ~N2  : MN2   =  4800.0 || ~N3  : MN3   =  6800.0

Xf=2.93e+01 Omega h^2=2.69e-03

# Channels which contribute to 1/(omega) more than 1%.
# Relative contributions in % are displayed
   23% ~etI ~etR ->nu3 nu3
   13% ~etR ~etR ->Wp Wm
   13% ~etI ~etI ->Wp Wm
   11% ~etR ~etR ->nu3 nu3
   11% ~etI ~etI ->nu3 nu3
    5% ~etI ~etR ->nu2 nu2
    4% ~etI ~etR ->nu2 nu3
    3% ~etR ~etR ->Z Z
    3% ~etI ~etI ->Z Z
    3% ~etR ~etR ->nu2 nu2
    3% ~etI ~etI ->nu2 nu2
    2% ~etR ~etR ->nu2 nu3
    2% ~etI ~etI ->nu2 nu3

==== Calculation of CDM-nucleons amplitudes =====
        TREE LEVEL
CDM-nucleon micrOMEGAs amplitudes:
proton:  SI  -1.254E-18  SD  0.000E+00
neutron: SI  -1.273E-18  SD  0.000E+00
        BOX DIAGRAMS
CDM-nucleon micrOMEGAs amplitudes:
proton:  SI  -1.254E-18  SD  0.000E+00
neutron: SI  -1.273E-18  SD  0.000E+00
CDM-nucleon cross sections[pb]:
 proton  SI 6.819E-28  SD 0.000E+00
 neutron SI 7.020E-28  SD 0.000E+00

======== Direct Detection ========
73Ge: Total number of events=1.45E-22 /day/kg
Number of events in 10 - 50 KeV region=7.85E-23 /day/kg
131Xe: Total number of events=2.42E-22 /day/kg
Number of events in 10 - 50 KeV region=1.22E-22 /day/kg
23Na: Total number of events=1.45E-23 /day/kg
Number of events in 10 - 50 KeV region=7.90E-24 /day/kg
I127: Total number of events=2.37E-22 /day/kg
Number of events in 10 - 50 KeV region=1.21E-22 /day/kg
```

We note that this scenario leads to a tiny dark matter relic density, of the order of $2.69 \cdot 10^{-3}$, due to the large annihilation cross-sections into pairs of neutrinos and gauge bosons (W^+W^- and ZZ). In fact, these are not only annihilations, but also co-annihilations with the η_R state, which is almost degenerate in mass. Regarding the calculation of the direct detection cross-sections, the most relevant information is given after `CDM-nucleon cross sections[pb]`. These are the (spin independent and dependent) cross-sections with proton and neutron in pb. It is usually convenient to multiply these values by a factor 10^{-36} to get the cross-sections in cm^2, the units commonly employed by the experimental collaborations. We find that in the **BS2** point these cross-sections are tiny.

Finally, `MicrOmegas` also computes the number of recoil events per day in the 10 - 50 keV region for a kg of ^{73}Ge, ^{131}Xe, ^{23}Na and ^{127}I. Again, and due to the small direct detection cross-sections, these numbers are tiny in the **BS2** point. The largest number of events would be obtained in ^{131}Xe, but even in this case we would expect only $\sim 10^{-22}\,\text{events}\,\text{kg}^{-1}\,\text{day}^{-1}$.

Before concluding the lecture, we emphasize that many other dark matter related observables can be computed using `MicrOmegas`. For a detailed list see the `MicrOmegas` manual [17].

3.6 Summary of the lecture

In this lecture we learned how to use `MicrOmegas` to compute observables related to dark matter physics. Since we had produced the input files with `SARAH`, we did not have to worry about how to write them. Instead, we focused on their practical use to obtain reliable predictions for the DM relic density and direct and indirect detection rates.