MicrOMEGAs Howto

These are some notes on installing and running the MicrOMEGAs program [1, 2] on a linux/unix platform. The notes are not meant to be complete. For details (and if you run into trouble) always refer to the official manuals [1, 2].

1 Installation

Download the latest version (2.0.6) from the MicrOMEGAs web site

```
http://wwwlapp.in2p3.fr/lapth/micromegas/
```

You will receive a tarred and zipped file `micromegas_2.0.6.tgz`. Unpack and untar this file

```
tar -zxvf micromegas_2.0.6.tgz
```

This will create the directory `micromegas_2.0.6` with several files and subdirectories. Now enter this directory:

```
cd micromegas_2.0.6
```

and compile the program: type

```
gmake
```

and keep your fingers crossed. If you get an error, consult the manual or the program website.

2 Supersymmetry

You should see that one of the subdirectories is called "MSSM". Now go to this subdirectory:

```
cd MSSM
```

You have the option to run either the model of minimal supergravity (MSUGRA), or the general MSSM with a multitude of parameters each individually specified at the weak scale.

2.1 General MSSM

To run the general MSSM, first create the corresponding executable using either the fortran

```
gmake main=omg.F
```

or C version of the source code

```
gmake main=omg.c
```

Either one should create an executable file called `omg`. If you attempt to execute this file:

```
./omg
```

the program will complain about lack of inputs. Those should be specified in a separate input data file which should be handed to the program. Sample input files are contained in the directory `data`. Each one specifies a set of variables and their values. For the meaning of the input variables, see the manual, although the notation is pretty suggestive. For example, to run the program with the inputs contained in the first data file, do

```
./omg data/data01.par
```

2.2 The MSUGRA model

In order to run MSUGRA, first create the corresponding executable `sugomg`:

```
gmake main=sugomg.F
```

or

```
gmake main=sugomg.c
```

If you attempt to run it, e.g.

```
./sugomg
```

the program again will complain about lack of inputs (the MSUGRA parameters), but this time it will also tell you their proper input sequence on the command line. From this point on just follow the instructions on the screen. For example, to run the MSUGRA point $m_0 = 100$ GeV, $M_{1/2} = 300$ GeV, $A_0 = 0$ and $\tan\beta = 20$, type

```
./sugomg 100 300 0 20
```

3 Implementing a New Model: the nMSM

Now you are ready to implement your own model. Our example is the new Minimal Standard Model (nMSM) of Ref. [3]. Go to the top micromegas directory and create a new project directory:

```
./newProject nMSM
```

which will create a new working directory `nMSM`. Go to this directory:

```
cd nMSM
```

It is a good idea to read the `README` file at this point which explains the conventions in defining new generic models.

The new model files need to be placed in the subdirectory `work/models`. Notice that at this point this subdirectory is empty. Therefore your first job is to create the relevant files in CalcHEP [4] format and place them in the directory `work/models`. Note that the model number should be number 1, meaning your model files should be named `extlib1.mdl, func1.mdl, lgrng1.mdl, prtcls1.mdl, vars1.mdl`.

Notice that calchep is available directly in your new project directory (in this case `nMSM`). To save typing, it is convenient to recycle existing calchep model files, preferably for models which are closest to yours. For example, you can start from the MSSM or the SM, and only add the new pieces. In our case it is convenient to start from the SM. Therefore, we should first put the usual SM files in the `work/models` directory, and later modify them with `calchep` executed from within the `work` directory.

3.1 Obtaining the SM model files

You can retrieve the SM model files for CalcHEP by simply downloading and unpacking CalcHEP - there is no need to compile it - the model files will already be in the directory `models`. Copy those to your `nMSM/work/models` directory. Now you can edit them by launching CalcHEP from within your `nMSM/work` directory:

```
./calchep &
```

3.2 Implementing the nMSM

You can first rename the Standard Model to “nMSM”, then add the S scalar to the particle list as `~S`. Then define its mass and width, and add its interactions in the Lagrangian. After saving your changes, as a sanity check, you may wish to check a few processes involving S

within CalcHEP.

3.3 Calculating the relic density in nMSM

Now go to the nMSM directory and create the executable for the relic density calculation:

```
gmake main=omg.c
```

This will create the executable `omg` which you should run with a file specifying the values of the parameters in your model, for example the mass and couplings of S, e.g.

```
./omg nMSMinput.par
```

This, however, will fail due to a bug. The patch is as follows. Go to the `nMSM/lib` directory and replace the Makefile there with the Makefile linked from the PiTP web page of the lectures. Then repeat from the beginning of this subsection 3.3.

4 Calculation of annihilation spectra

The indirect detection rates of dark matter particles in the galactic halo through their decay products in photons, positrons or antiprotons depend on the dark matter annihilation cross-sections at small relative velocity. The cross-sections for the different 2-body annihilation channels of the LOP are calculated automatically in any model. The program then provides the continuum spectrum for γ, e^+, $\bar{p}$, ν production.

First, make the corresponding executable:

```
gmake main=spectrum.c
```

and then to run the MSUGRA point above ($m_0 = 100$ GeV, $M_{1/2} = 300$ GeV, $A_0 = 0$ and $\tan\beta = 20$) try

```
./spectrum 100 300 0 20
```

and see what happens.

Good luck!

A MSSM parameter list

A.1 Weak scale parameters

When running MicrOMEGAs, the input file must specify the set of weak scale SUSY parameters as in Table 1. The sfermion masses correspond to the soft masses without the D-term.

Table 1: Weak scale input parameters in the notation of CompHEP

mu	μ Higgs mass parameter
MG1	M_1 Gaugino mass
MG2	M_2 Gaugino mass
MG3	M_3 Gaugino mass
Ml1	Left-handed slepton mass, 1st generation, $m_{\tilde{L}_{1L}}$
Ml2	Left-handed slepton mass, 2nd generation, $m_{\tilde{L}_{2L}}$
Ml3	Left-handed slepton mass, 3rd generation, $m_{\tilde{L}_{3L}}$
Mr1	Right-handed selectron mass, $m_{\tilde{e}_R}$
Mr2	Right-handed smuon mass, $m_{\tilde{\mu}_R}$
Mr3	Right-handed stau mass, $m_{\tilde{\tau}_R}$
Mq1	Left-handed squark mass, 1st generation, $m_{\tilde{Q}_{1L}}$
Mq2	Left-handed squark mass, 1st generation, $m_{\tilde{Q}_{2L}}$
Mq3	Left-handed squark mass, 1st generation, $m_{\tilde{Q}_{3L}}$
Mu1	Right-handed u-squark mass, $m_{\tilde{u}_R}$
Mu2	Right-handed c-squark mass, $m_{\tilde{c}_R}$
Mu3	Right-handed t-squark mass, $m_{\tilde{t}_R}$
Md1	Right-handed d-squark mass, 1st generation$m_{\tilde{d}_R}$
Md2	Right-handed s-squark mass, $m_{\tilde{s}_R}$
Md3	Right-handed b-squark mass, $m_{\tilde{b}_R}$
Atop	Trilinear coupling for top quark, A_t
Ab	Trilinear coupling for b-quark, A_b
Atau	Trilinear coupling for τ-lepton, A_τ
Am	Trilinear coupling for μ, A_μ
MH3	Mass of Pseudoscalar Higgs (M_A)
tb	Ratio of the vacuum expectation values of the scalar doublets

A.2 SUGRA GUT-scale parameters

The set of input parameters for the SUGRA model are the usual

```
m0      common scalar mass at GUT scale
mhf     common gaugino mass at GUT scale
a0      trilinear soft breaking parameter at GUT scale
tb      tan(beta) or the  ratio of vacuum expectation values
sgn     +/-1,  sign of Higgsino mass term
mtop    pole mass of the top quark
model    = 1  SUGRA model
         = 2  SUGRA with true gauge coupling unification at the GUT scale
```

B Specific requirements for implementing new models

This part is copied from Section 5.3 in [2].

The general format to be used for model files is described in [4]. These are only a few specific points needed for `MicrOMEGAs`.

Names of odd particles. The name of an odd particles must start with `~`. With this convention, automatic identification of the R-parity odd particles is done by `MicrOMEGAs`. For the purpose of optimizing the code, it is recommended that the first odd particle in the `models/prtcls1.mdl` list be a potential LOP candidate.

Masses of odd particles. A `*` symbol should be added before the masses of R-parity odd particles that are not independent parameters of the model, that is the ones that are found in the `models/func1.mdl` file. This is to force the inclusion of the corresponding parameter in any generated code.

Widths of odd particles. The widths of odd particles should be independent parameters of the model. All odd particles, including the LOP, must have a width in order to avoid divergences in cross-sections. For example, in coannihilation processes with exchange of the LOP in t-channel, one can meet a pole if the width of the LOP is zero. The width of the LOP is set within `MicrOMEGAs` to $M_{LOP}/100$, see explanations in [1].

Automatic calculation of widths. In previous `CalcHEP` versions the widths of particles were treated as independent parameters. Starting from CalcHEP_2.4 there is an option to calculate widths automatically. To switch on the mechanism of automatic width calculation one must add the '!' symbol in front of the width in the particle list (`models/prtcls1.mdl`). If necessary, the corresponding parameter should be removed from the list of independent parameters (`models/vars1.mdl`) and/or from the list of constrained parameters (`models/func1.mdl`).

For the relic density calculation, this trick is used for widths of Higgs particles which occur as s-channel resonances.

References

[1] G. Belanger, F. Boudjema, A. Pukhov and A. Semenov, “MicrOMEGAs: Version 1.3,” Comput. Phys. Commun. **174**, 577 (2006) [arXiv:hep-ph/0405253].

[2] G. Belanger, F. Boudjema, A. Pukhov and A. Semenov, “micrOMEGAs2.0: A program to calculate the relic density of dark matter in a generic model,” Comput. Phys. Commun. **176**, 367 (2007) [arXiv:hep-ph/0607059].

[3] H. Davoudiasl, R. Kitano, T. Li and H. Murayama, “The new minimal standard model,” Phys. Lett. B **609**, 117 (2005) [arXiv:hep-ph/0405097].

[4] A. Pukhov, “CalcHEP 3.2: MSSM, structure functions, event generation, batchs, and generation of matrix elements for other packages,” arXiv:hep-ph/0412191.