



# MicrOMEGAs: hands-on session

ISAPP School  
Gamma rays to shed light on dark matter



Andreas Goudelis  
LPC - Clermont Ferrand

# Outline

- Introduction: what is micrOMEGAs ?
- Implementing a new model : singlet scalar dark matter
- Importing a model in micrOMEGAs
- Running micrOMEGAs: Freeze-out vs freeze-in

# A bit of history: DM abundance

In the early 2000's, if you wanted to compute the dark matter abundance predicted by your favourite model, you had to:

- Write down your Lagrangian.
- Extract all vertices.
- Figure out which processes are relevant for dark matter production/depletion.
- Compute all the cross-sections.
- Write down the relevant Boltzmann equations.
- Code all these expressions and numerically solve your Boltzmann equations.

How complicated is it?

# A bit of history: DM abundance

In the early 2000's, if you wanted to compute the dark matter abundance predicted by your favourite model, you had to:

- Write down your Lagrangian.
- Extract all vertices.
- Figure out which processes are relevant for dark matter production/depletion.
- Compute all the cross-sections.
- Write down the relevant Boltzmann equations.
- Code all these expressions and numerically solve your Boltzmann equations.

How complicated is it?

My take: I wasn't in research at the time, but to me it sounds pretty complicated!

# A bit of history: the MSSM

At the time, supersymmetry was still the name of the game. Consider the Minimal Supersymmetric Standard Model (MSSM):

- 28 new particles (+ 17 from the SM).
- All of them can, in principle, coannihilate (either with the dark matter candidate or among them).
- In total, this amounts to more than 2800 processes (*in principle*, although typically not all are relevant, but still...).

This is *a lot* of work!

# A bit of history: the MSSM

At the time, supersymmetry was still the name of the game. Consider the Minimal Supersymmetric Standard Model (MSSM):

- 28 new particles (+ 17 from the SM).
- All of them can, in principle, coannihilate (either with the dark matter candidate or among them).
- In total, this amounts to more than 2800 processes (*in principle*, although typically not all are relevant, but still...).

This is *a lot* of work!

People *did* do it and even developed public codes (*cf* also T. Bringmann's lecture).

→ Common feature: all relevant expressions were *hard-coded*.



This procedure had to be repeated for every new model

# A bit of history: automation

As usual, the idea of automatising some steps of this procedure arose. Remember our list of tasks:

- Write down your Lagrangian.
- Extract all vertices.
- Figure out which processes are relevant for dark matter production/depletion.
- Compute all the cross-sections.
- Write down the relevant Boltzmann equations.
- Code all these expressions and numerically solve your Boltzmann equations.

# A bit of history: automation

As usual, the idea of automatising some steps of this procedure arose. Remember our list of tasks:

- Write down your Lagrangian.
- Extract all vertices.
- Figure out which processes are relevant for dark matter production/depletion.
- Compute all the cross-sections.
- Write down the relevant Boltzmann equations.
- Code all these expressions and numerically solve your Boltzmann equations.

these tend to be the most time-consuming *and prone to human error*.



# A bit of history: automation

As usual, the idea of automatising some steps of this procedure arose. Remember our list of tasks:

- Write down your Lagrangian.
- Extract all vertices.
- Figure out which processes are relevant for dark matter production/depletion.
- Compute all the cross-sections.
- Write down the relevant Boltzmann equations.
- Code all these expressions and numerically solve your Boltzmann equations.

Today some of these steps can be automated using different tools. But

- i)* These tools either didn't exist at the time or were less developed.
- ii)* You still need to do quite a bit of coding to bring everything together.

Can we make our lives a bit easier?

# A bit of history: micrOMEGAs 1

Original idea behind micrOMEGAs: once a particle content and their interactions are specified, the rest can be automated. In particular:

- Figure out which processes are relevant for dark matter production/depletion.
- Compute all the cross-sections.
- Write down the relevant Boltzmann equations.
- Code all these expressions and numerically solve your Boltzmann equations.

micrOMEGAs: A program for calculating the relic density in the MSSM

G. Bélanger<sup>1</sup>, F. Boudjema<sup>1</sup>, A. Pukhov<sup>2</sup>, A. Semenov<sup>1</sup>

[arXiv:hep-ph/0112278](https://arxiv.org/abs/hep-ph/0112278)

→ All cross-sections computed at tree-level with CompHEP/CalcHEP.

<https://theory.sinp.msu.ru/~pukhov/calchep.html>

# So, what is micrOMEGAs?

MicrOMEGAs is a computer program (in C) for the automatic computation of dark matter observables in *generic* extensions of the Standard Model. It can:

- Compute the predicted dark matter abundance in a user-defined model according to the freeze-out or freeze-in pictures (incl. two-component DM scenarios).
- Compute observables for direct detection of dark matter.
- Compute observables for indirect detection of dark matter.
- Compute decay widths and cross-sections (*e.g.* for collider analyses).
- For some models (*e.g.* MSSM/NMSSM) compute observables such as  $b \rightarrow s \gamma$ .
- Interface to other PHENO pieces of software (*e.g.* SModelS, HiggsBounds), plot useful quantities etc.

} Under different  
astro assumptions

**micrOMEGAs5.0 : freeze-in**

Current version:

G. Bélanger<sup>1†</sup>, F. Boudjema<sup>1‡</sup>, A. Goudelis<sup>2§</sup>, A. Pukhov<sup>3¶</sup>, B. Zaldivar<sup>1††</sup>

arXiv:1801.03509

<https://lapth.cnrs.fr/micromegas/>

# How to use micrOMEGAs

The basic steps to follow if you want to use micrOMEGAs to compute observables for your favourite dark matter model are:

Write down your Lagrangian

Extract all vertices and code them in an appropriate form

Edit a script in C

Have fun!

This can be a bit tedious too...

# How to use micrOMEGAs - bis

Actually, the task of writing down the Feynman rules has also been automated thanks to programs like LanHEP, FeynRules or SARAH. So in practice:

Write down your your Lagrangian  
and code it in an appropriate form

Generate CalcHEP model files  
and import them in micrOMEGAs

Edit a script in C

Have fun!

# How to use micrOMEGAs - bis

Actually, the task of writing down the Feynman rules has also been automated thanks to programs like LanHEP, FeynRules or SARAH. So in practice:

Write down your your Lagrangian and code it in an appropriate form

Generate CalcHEP model files and import them in micrOMEGAs

Edit a script in C

Have fun!

For more information on LH/FR/SARAH please visit

- <https://theory.sinp.msu.ru/~semenov/lanhep.html>
- <https://feynrules.irmp.ucl.ac.be/>
- <https://sarah.hepforge.org/>

Goal of this session:  
Go through all of these steps together

# Step 1: Implementing a new model

NB: Strictly speaking,  
not yet in micrOMEGAs

# Example model: singlet scalar DM

Consider an extension of the Standard Model by a real singlet scalar field  $s$  which is odd under a discrete  $\mathbf{Z}_2$  symmetry, with all the SM particles being even.

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2}(\partial_\mu s)(\partial^\mu s) - \frac{\mu_s^2}{2}s^2 - \frac{\lambda_s}{4}s^4 - \frac{\lambda_{sh}}{2}s^2(H^\dagger H)$$

Once EW symmetry is broken,  $s$  can interact (at tree-level and max  $2 \leftrightarrow 2$ ):

- In pairs, with a single Higgs boson ( $s$ - $s$ - $h$  vertex).
- In pairs, with a pair of Higgs bosons ( $s$ - $s$ - $h$ - $h$  vertex).

In order to use micrOMEGAs we need to specify (in CalcHEP format):

- The nature of the particles in the model (how many particles, their names, spins and  $\mathbf{Z}_2$  charges).
- How they interact with each other (the vertices).

→ This can be done by directly writing this information down directly in CalcHEP formal. Thankfully, there is a simpler way.



# Model file generators

Programs such as LanHEP, FeynRules or SARAH allow us to specify our particle content and input a Lagrangian in a fairly intuitive way and obtain as an output the corresponding information that is needed by CalcHEP/micrOMEGAs, in appropriate format (“model file”).

Each of these programs has its vices and virtues and can do much more than what we'll see today, for more info check out their webpages and manuals!

Examples of what they can do include:

- Generate counterterms for one-loop calculations.
- Calculate and write numerical code for renormalization group equations.
- Compute decay widths.
- Generate model files for other software such as MadGraph, SPheno etc.

→ In the end, which program you'll use depends on what you want to do and which one you find more convenient.

Today we'll be using FeynRules, which is based on Mathematica. Variations among programs do exist, but the general idea is similar.

NB: LanHEP is included by default in MO

# Writing down a model file: preliminaries


Let's start by writing down a FeynRules model file for the singlet scalar DM model.

1) The model contains one particle other than the Standard Model ones:  $s$ .

2) As we discussed, the imposition of the discrete  $\mathbf{Z}_2$  symmetry restricts the allowed terms to:

$$\mathcal{L} = \mathcal{L}_{\text{SM}} + \frac{1}{2}(\partial_\mu s)(\partial^\mu s) - \frac{\mu_s^2}{2}s^2 - \frac{\lambda_s}{4}s^4 - \frac{\lambda_{sh}}{2}s^2(H^\dagger H)$$

3) The model is fully described by the following parameters :

- The physical DM mass  $m_s$
  - The portal couplings  $\lambda_{sh}$
  - The quartic coupling  $\lambda_s$
- 
- “External parameters”

whereas  $\mu_s$  can be computed from the rest. } → “Internal parameter”

NB: Of course, the model *also* involves the SM particles, Lagrangian and parameters, which *also* need to be specified!

# Some elements of a FeynRules model file

Declaring parameters:

```
muss == {  
    TeX->Subscript[\[Lambda],s],  
    ComplexParameter -> False,  
    ParameterType -> Internal,  
    Value -> Sqrt[mss^2 - lamssh*vev^2],  
    Description->"Dark scalar mass parameter"},  
  
lamssh == {  
    TeX->Subscript[\[Lambda],sh],  
    ComplexParameter -> False,  
    ParameterType -> External,  
    Value -> 0.1,  
    InteractionOrder->{QED, 2},  
    Description->"Dark scalar - Higgs doublet interaction"}
```

# Some elements of a FeynRules model file

Declaring parameters:

```
muss == {  
  TeX->Subscript[ $\lambda$ ,s],  
  ComplexParameter -> False,  
  ParameterType -> Internal,  
  Value -> Sqrt[mss^2 - lamssh*vev^2],  
  Description->"Dark scalar mass parameter"},  
  
lamssh == {  
  TeX->Subscript[ $\lambda$ ,sh],  
  ComplexParameter -> False,  
  ParameterType -> External,  
  Value -> 0.1,  
  InteractionOrder->{QED, 2},  
  Description->"Dark scalar - Higgs doublet interaction"}
```

A bit of offline  
work is required.



# Some elements of a FeynRules model file

Declaring parameters:

```
muss == {
  TeX->Subscript[\[Lambda],s],
  ComplexParameter -> False,
  ParameterType -> Internal,
  Value -> Sqrt[mss^2 - lamssh*vev^2],
  Description->"Dark scalar mass parameter"},

lamssh == {
  TeX->Subscript[\[Lambda],sh],
  ComplexParameter -> False,
  ParameterType -> External,
  Value -> 0.1,
  InteractionOrder->{QED, 2},
  Description->"Dark scalar - Higgs doublet interaction"}
```

A bit of offline work is required.

Declaring particles:

```
S[6] == {
  ClassName      -> ss,
  SelfConjugate  -> True,
  QuantumNumbers -> {},
  Mass           -> {mss, 10.},
  Width          -> {wss, 0.},
  ParticleName   -> "~ss",
  PDG            -> 254,
  PropagatorLabel -> "ss",
  PropagatorType -> ScalarDash,
  ParticleName   -> {"~ss"},
  FullName       -> {"Dark scalar" }
```

# Some elements of a FeynRules model file

Declaring parameters:

```
muss == {
  TeX->Subscript[\[Lambda],s],
  ComplexParameter -> False,
  ParameterType -> Internal,
  Value -> Sqrt[mss^2 - lamssh*vev^2],
  Description->"Dark scalar mass parameter"},

lamssh == {
  TeX->Subscript[\[Lambda],sh],
  ComplexParameter -> False,
  ParameterType -> External,
  Value -> 0.1,
  InteractionOrder->{QED, 2},
  Description->"Dark scalar - Higgs doublet interaction"}
```

A bit of offline work is required.

Declaring particles:

```
S[6] == {
  ClassName      -> ss,
  SelfConjugate  -> True,
  QuantumNumbers -> {},
  Mass           -> {mss, 10.},
  Width          -> {wss, 0.},
  ParticleName   -> "~ss",
  PDG            -> 254,
  PropagatorLabel -> "ss",
  PropagatorType -> ScalarDash,
  ParticleName   -> {"~ss"},
  FullName       -> {"Dark scalar"} }
```

Note the tilde: it is crucial for micrOMEGAs to understand that this is a  $\mathbb{Z}_2$ -odd particle, hence, a DM candidate!

# Some elements of a FeynRules model file

Declaring parameters:

```
muss == {
  TeX->Subscript[\[Lambda],s],
  ComplexParameter -> False,
  ParameterType -> Internal,
  Value -> Sqrt[mss^2 - lamssh*vev^2],
  Description->"Dark scalar mass parameter"},

lamssh == {
  TeX->Subscript[\[Lambda],sh],
  ComplexParameter -> False,
  ParameterType -> External,
  Value -> 0.1,
  InteractionOrder->{QED, 2},
  Description->"Dark scalar - Higgs doublet interaction"}
```

A bit of offline work is required.

Declaring particles:

```
S[6] == {
  ClassName      -> ss,
  SelfConjugate  -> True,
  QuantumNumbers -> {},
  Mass           -> {mss, 10.},
  Width          -> {wss, 0.},
  ParticleName   -> "~ss",
  PDG            -> 254,
  PropagatorLabel -> "ss",
  PropagatorType -> ScalarDash,
  ParticleName   -> {"~ss"},
  FullName       -> {"Dark scalar" } }
```

Note the tilde: it is crucial for micrOMEGAS to understand that this is a  $Z_2$ -odd particle, hence, a DM candidate!

Writing down the Lagrangian:

```
LNP := Block[{i},
  feynmangaugerules = If[Not[FeynmanGauge], {G0|GP|GPbar ->0}, {}];
  ExpandIndices[
    del[ss,mu]^2 - muss^2/2 ss^2 - lamssh/4 ss^4 -
    lamssh ss^2 Phibar[i] Phi[i],
    FlavorExpand->{SU2D}]/.feynmangaugerules
  ];
LTOT := LSM + LNP;
```

# Creating model files

Please go to:

`~/feynrules-current/Models/SingletScalarDM/`



## Step 2: Importing our model in micrOMEGAs

# Importing a new model

The first step is to create a dedicated directory structure for our model. This is done with

```
./newProject SingletScalarDM/
```

The model files that we generated with FeynRules are to be placed in

```
~/micromegas_5.2.7.a/SingletScalarDM/work/models/
```

and then it's always a good idea to go to

```
~/micromegas_5.2.7.a/SingletScalarDM/work/
```

run: `./calchep` , check the model and make sure that “The model is Ok”.

Now let's have a look at our model files

# Elements of a micrOMEGAs model

Each model is described by a basic set of files:

- An external library file (extlib1.mdl - not used today)

# Elements of a micrOMEGAs model

Each model is described by a basic set of files:

- An external library file (extlib1.mdl - not used today)
- A file describing all the particles of the model (prtcls1.mdl)

```
SingletScalarDM
Particles
Full name |A          |A+          | number |2*spin| mass          |width          |color|aux|>LaTeX(A)<|>LaTeX(A+) <|
Photon    |a          |a           |22      |2     |0             |0              |1    |G  |a           |a           |
Z         |Z          |Z           |23      |2     |MZ            |!WZ            |1    |G  |Z           |Z           |
```

# Elements of a micrOMEGAs model

Each model is described by a basic set of files:

- An external library file (extlib1.mdl - not used today)
- A file describing all the particles of the model (prtcls1.mdl)

```
SingletScalarDM
Particles
Full name |A          |A+          | number |2*spin| mass          |width          |color|aux|>LaTeX(A)<|>LaTeX(A+) <|
Photon    |a          |a           |22      |2     |0             |0              |1    |G  |a          |a          |
Z         |Z          |Z           |23      |2     |MZ           |!WZ           |1    |G  |Z          |Z          |
```

- A file describing all the external parameters of the model (vars1.mdl)

```
lamssh          |0.1          |Dark scalar - Higgs doublet interaction          |
```

# Elements of a micrOMEGAs model

Each model is described by a basic set of files:

- An external library file (extlib1.mdl - not used today)
- A file describing all the particles of the model (prtcls1.mdl)

```
SingletScalarDM
Particles
Full name |A          |A+          | number |2*spin| mass          |width          |color|aux|>LaTeX(A)<|>LaTeX(A+) <|
Photon    |a          |a          | 22     | 2     | 0            | 0             | 1   |G  |a          |a          |
Z         |Z          |Z          | 23     | 2     | MZ           | !WZ           | 1   |G  |Z          |Z          |
```

- A file describing all the external parameters of the model (vars1.mdl)

```
lamssh          |0.1          |Dark scalar - Higgs doublet interaction          |
```

- A file containing all the vertices of the model (lgrng1.mdl)

```
SingletScalarDM
Vertices
A1          |A2          |A3          |A4          |>          |Factor          |<|> Lorentz part          |<|
H           |H           |H           |H           |x1x0       |                |1
```

# Elements of a micrOMEGAs model

Each model is described by a basic set of files:

- An external library file (extlib1.mdl - not used today)
- A file describing all the particles of the model (prtcls1.mdl)

```
SingletScalarDM
Particles
Full name |A          |A+          | number |2*spin| mass          |width          |color|aux|>LaTeX(A)<|>LaTeX(A+) <|
Photon    |a          |a          | 22     | 2     | 0            | 0             | 1   |G  |a          |a          |
Z         |Z         |Z         | 23     | 2     | MZ          | !WZ          | 1   |G  |Z          |Z          |
```

- A file describing all the external parameters of the model (vars1.mdl)

```
lamssh          |0.1          |Dark scalar - Higgs doublet interaction          |
```

- A file containing all the vertices of the model (lgrng1.mdl)

```
SingletScalarDM
Vertices
A1          |A2          |A3          |A4          |>          |Factor          |<|> Lorentz part          |<|
H           |H           |H           |H           |x1x0       |                |1
```

- A file containing all the internal parameters of the model (func1.mdl)

```
muss          |pow(pow(mss,2)-lamssh*pow(vev,2),0.5)          |% Dark scalar mass parameter
x1x0          |-6*lam          |% H H H H coupling.
```

# Elements of a micrOMEGAs model

Each model is described by a basic set of files:

- An external library file (func1.mdl)

- A file describing the particles

Full name	Spin	Charge
Photon	1	0
Z	0	0

You can, of course, write all of this by hand.

You just need to respect the CalcHEP file format.

I find it easier to use an external program.

- A file describing the vertices

Full name	Spin	Charge
lamssh	0	0

- A file containing all the internal model (func1.mdl)

Full name	Spin	Charge
A1	0	0
A2	0	0
A3	0	0
H	0	0

- A file containing all the internal model (func1.mdl)

Full name	Spin	Charge
muss	0	0
x1x0	0	0



# A closer look into our model files

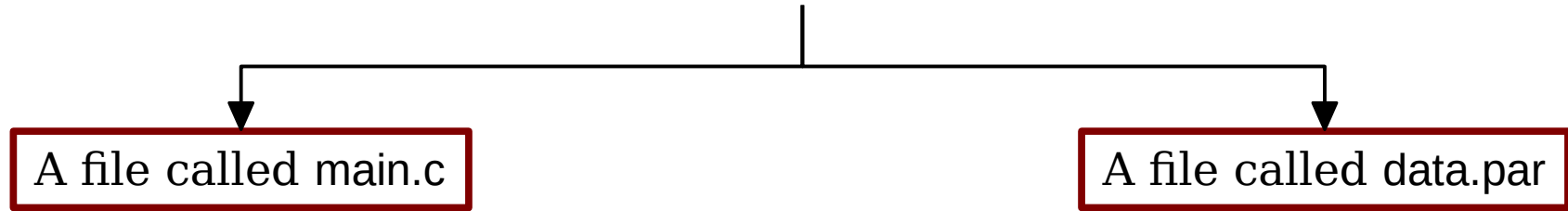
Please go to:

`~/micromegas_5.2.7.a/SingletScalarDM/work/models/`

Step 3: Having fun!

# The model directory

The directory `~/micromegas_5.2.7.a/SingletScalarDM/` that was created once you ran `./newProject SingletScalarDM/` should contain, among other things:

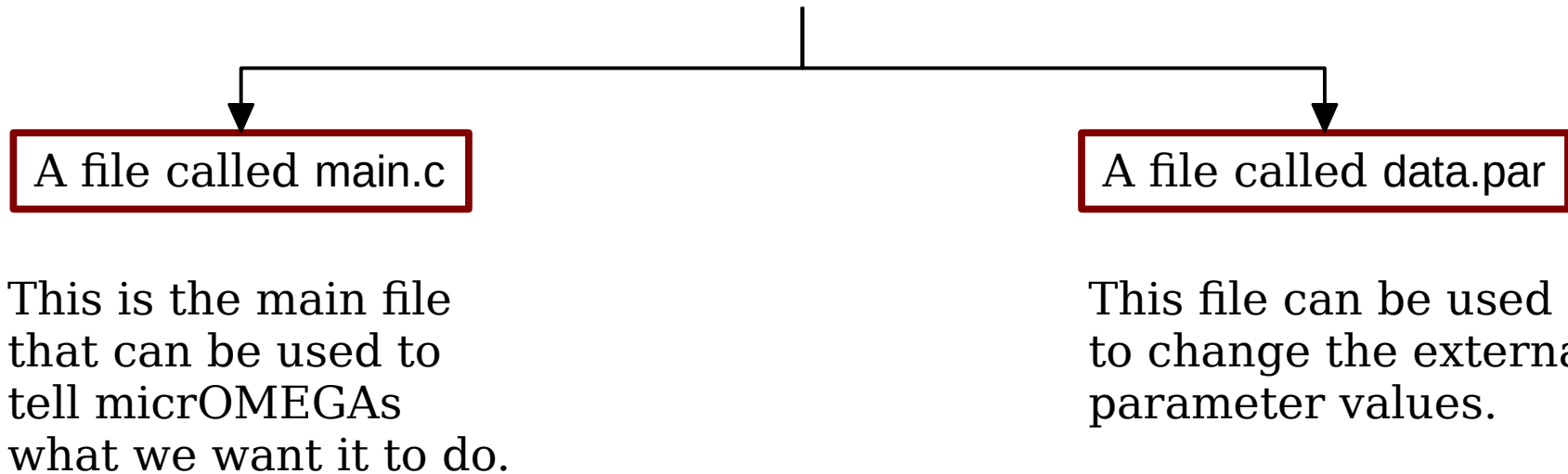


This is the main file that can be used to tell micrOMEGAS what we want it to do.

This file can be used to change the external parameter values.

# The model directory

The directory `~/micromegas_5.2.7.a/SingletScalarDM/` that was created once you ran `./newProject SingletScalarDM/` should contain, among other things:



In order to run the program, just execute:

```
make main=main.c
```

followed by:

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

Quite a few things should appear, which we'll discuss...

# The data.par file

Through this file we can change the value of any parameter that are listed in the (vars1.mdl) file.

· If this file is empty, the default values (listed in vars1.mdl) are used. In our case, we had set (already in FeynRules) as default values:

mss = 10 GeV

lamssh = 0.1



Q1: What is the main annihilation channel?  
Q2: Which other coupling is involved?

and we obtain a relic abundance  $\Omega h^2 \sim 0.3$ .

· In order to change the values of some parameters, let's edit the data.par file by adding the lines (one-by-one):

lamssh 0.16

mss 300

and rerunning the program each time.

Q: What do you expect to happen in each case?

# The main.c file: general structure

The default main.c file contains:

- Switches to turn on/off certain modules (functionalities).
- Commands to read the input data.par file and perform various checks.
- The commands

```
sortOddParticles(cdmName)
```

which sorts the  $\mathbf{Z}_2$ -odd particles with increasing masses and stores the names of the DM particles in CDM1, CDM2 and

```
qNumbers(CDM1, &spin2, &charge3, &cdim)
```

which returns the quantum numbers of the particle CDM1.

- Various modules computing different quantities (relic density, DM-nucleon scattering cross-section etc), plotting etc.

→ This file can be edited to suit what exactly it is that you want to achieve.

Today we'll only go through a few of these functionalities. For a full account, please *cf* the micrOMEGAs manual in `~/micromegas_5.2.7.a/man/`

# A closer look into the main.c file

Please *cf*:

`~/micromegas_5.2.7.a/SingletScalarDM/main.c`

# Writing a simple program

As a final exercise, let's modify the default main.c file to write a simple program to:

- Change the values of the DM mass and its coupling from within the code. This can be done with `assignValW("var", val)`, which allows one to change the value of parameter `var` from within the main script.
- Compute the freeze-out DM abundance and print the contributions of the main annihilation channels.
- Compute the spin-independent and spin-dependent WIMP-proton/neutron scattering cross-sections.
- Change the DM coupling to a value which is more appropriate for freeze-in.
- Compute the total freeze-in DM abundance, as well as the contribution from Higgs decays.



# Writing a simple program

As a final exercise, let's modify the default main.c file to write a simple program to:

- Change the values of the DM mass and its coupling from within the code. This can be done with `assignValW("var", val)`, which allows one to change the value of parameter `var` from within the main script.
- Compute the freeze-out DM abundance and print the contributions of the main annihilation channels.
- Compute the spin-independent and spin-dependent WIMP-proton/neutron scattering cross-sections.
- Change the DM coupling to a value which is more appropriate for freeze-in.
- Compute the total freeze-in DM abundance, as well as the contribution from Higgs decays.

MicrOMEGAs does *not* check whether freeze-out/freeze-in conditions apply to your model and/or parameter values. You must *never* use a program as a black box, you must have at least some knowledge of what you're doing.

You can also use the script `mymain.c` for guidance, but it's better to first give it a shot yourselves!

# Summary and final comments

- micrOMEGAs is a computer program that allows you to compute the dark matter abundance in the Universe along with numerous other observables in general extensions of the Standard Model of particle physics.
- Today we only had time to scratch the surface. You are invited to consult the manual to explore the full range of functionalities offered by micrOMEGAs.
- There are other dark matter codes, *e.g.* DarkSUSY (cf tomorrow's lecture by T. Bringmann) and MadDM, *and this is a good thing!* It allows for comparison, validation and versatility.
- Additional specialized tasks can be performed by other codes (*cf* the other hands-on sessions).
- Two of the strongest points of micrOMEGAs: capability to implement new models without modifying the core of the code (also shared by MadDM), possibility to compute the freeze-in DM abundance.
- Many ideas on how to further expand the code, also depending on requests from the community :) .

Thank you!

# Appendix

# Step-by-step summary

For convenience, let's summarize here the various steps that we took during the lecture in order to reach from the Lagrangian of the Singlet Scalar DM model to running micrOMEGAs:

1) Download and unpack FeynRules.

2) Go to

`~/feynrules-current/Models/`

and create a new folder, say

`SingletScalarDM/`

This will serve as our FeynRules workspace for the Singlet Scalar DM model.

3) Place the files `SingletScalarDM.fr`, `SM.fr`, `loadSingletScalarDM.nb` and `DiagonalCKM.rst` that you can find in the compressed file `SingletScalarDM.zip` in

`~/feynrules-current/Models/SingletScalarDM/`

The two `.fr` files contain our model (particles, Lagrangian, parameters etc), the Mathematica notebook (`.nb`) can be used to load the model and generate output and the `.rst` file is just a restriction to ignore CKM mixing.

# Step-by-step summary

4) Run the notebook loadSingletScalarDM.nb. Remember to adapt all appropriate paths as necessary according to your folder structure. The last command (WriteCHOutput[LTOT]) creates the desired model files for CalcHEP/micrOMEGAs (.mdl) and places them in the folder:

```
~/feynrules-current/Models/SingletScalarDM/SingletScalarDM-CH/
```

5) Download and unpack micrOMEGAs.

6) Go in

```
~/micromegas_5.2.7.a/
```

and run: make.

7) Create your workspace for the Singlet Scalar DM model by running

```
./newProject SingletScalarDM/
```

This should make the folder SingletScalarDM/ appear in ~/micromegas\_5.2.7.a/ .

8) Copy your CalcHEP/micrOMEGAs model files (.mdl) into

```
~/micromegas_5.2.7.a/SingletScalarDM/work/models/
```

Your model is ready!

# Step-by-step summary

9) The command `./newProject SingletScalarDM/` created, among other things, a default `main.c` and `data.par` file. In order to run the default main program with default parameter values (the latter were set already in FeynRules), move back to

```
~/micromegas_5.2.7.a/SingletScalarDM/
```

and execute

```
make main=main.c
```

Followed by

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

10) You can now edit `main.c` in order to add/subtract functionalities of micrOMEGAs that you wish to employ (*cf* also the example `mymain.c` that you downloaded from the ISAPP webpage, which actually does *less* than the default main file!). The full list can be found in the micrOMEGAs manual and publications. The file `data.par` can be used to change the values of the model parameters. This can also be done from within your main file, *cf* the example `mymain.c` (executed through `make main=mymain.c` followed by `./mymain`).

11) As a final comment, don't forget that steps 1-4 can also be performed with the help of LanHEP (which is included by default in micrOMEGAs) or SARAH.