MicrOMEGAs: hands-on session

Dark Tools 2025 workshop

Andreas Goudelis

LPC – Clermont Ferrand

Introductory comments

MicrOMEGAs has been under constant development since ~two decades. It incorporates numerous functionalities, too many to cover in an hour-long tutorial.

In this tutorial we will use MicrOMEGAs mainly in order to compute the dark matter relic abundance in a simple dark matter model, according to two different thermodynamical mechanisms.

Our case study will be the Singlet Scalar dark matter model:

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

The model is fully described by the following BSM parameters :

· The physical DM mass m_s (mss)

· The portal couplings λ_{sh} (lamssh)

· The quartic coupling λ_s

→ "External parameters"

whereas μ_s can be computed from the rest. → "Internal parameter"

We will assume that the User has implemented the model in CalcHEP format, through the means of his/her choice (*cf e.g.* talks on LANHEP and FeynRules).

Creating 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 (.mdl) are to be placed in

~/micromegas_6.2.4/SingletScalarDM/work/models/

At the stage we can have a look at the model files

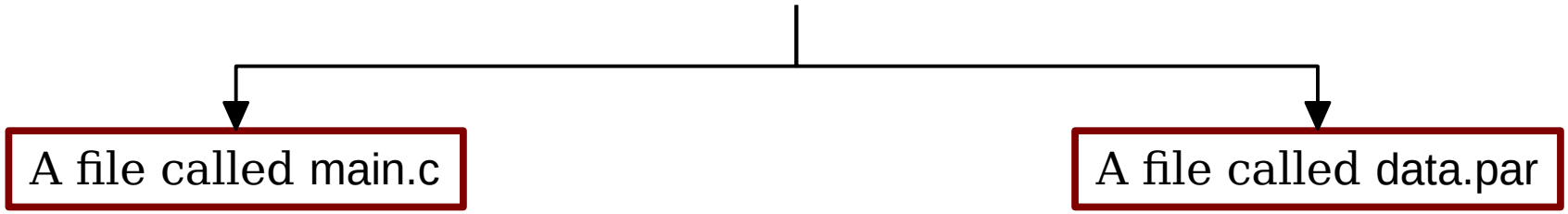
It’s also always a good idea to go to

~/micromegas_6.2.4/SingletScalarDM/work/

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

The model directory

The directory ~/micromegas_6.2.4/SingletScalarDM/ that was created once you ran ./newProject SingletScalarDM/ should contain, among other things:

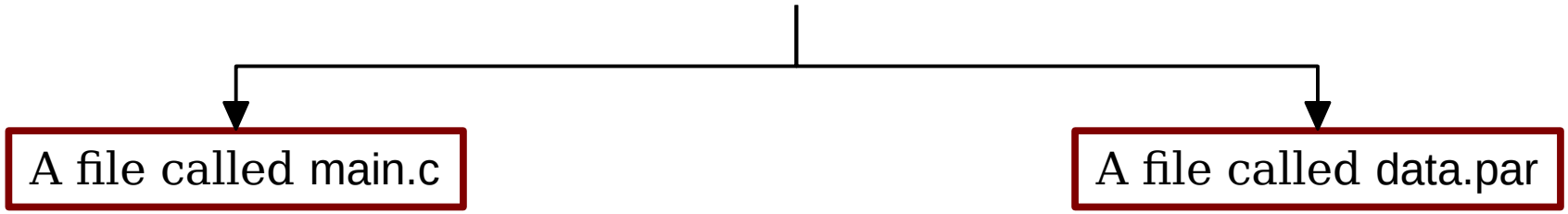


This is the main file that can be used to tell micrOMEGAs what we want it to do.

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

The model directory

The directory ~/micromegas_6.2.4/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.

In order to run the program, just execute:

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

followed by:

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

Quite a few things should appear in your terminal, some of 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: Which other coupling is involved?
Q2: Which is the main annihilation channel?

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 line:

lamssh 0.16

and reruning ./main data.par

Q: What do you expect to happen?

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 CDM[1], CDM[2],... and

```
qNumbers(CDM[1], &spin2, &charge3, &cdim)
```

which returns the quantum numbers of the particle CDM[1].

· 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_6.2.4/man/

Writing a simple program

Please *cf*:

~/micromegas_6.2.4/SingletScalarDM/mymain.c

Writing a simple program

Let’s have a look at a simplified main file (mymain.c). We will use it to:

· Change the values of the DM mass and its coupling. 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.

· Compute the freeze-in DM abundance and print the contributions of the main annihilation channels.

· Compute the DM abundance according to the “full” integrated Boltzmann equation and compare to the FO/FI limits.

→ In passing, we’ll also comment on some important topics

Initializations

```
/*====== Modules ===============
   Keys to switch on
   various modules of micrOMEGAs
================================*/

//...Calculate Freeze out relic density and display contribution of  individual channels
#define OMEGA
//...Calculate relic density in Freeze-in scenario
#define FREEZEIN
//...Calculate N dm relic density
#define NdmParticles

//...Calculate amplitudes and cross-sections for CDM-nucleon scattering
#define CDM_NUCLEON

/*===== end of Modules  ======*/

/*===== End of DEFINE  settings ===== */

#include"../include/micromegas.h"
#include"../include/micromegas_aux.h"
#include"lib/pmodel.h"

//=========================================================//
//=========================================================//
//=========================================================//
int main(int argc,char** argv)
{  int err;
   char cdmName[10];
   int spin2, charge3,cdim;
   int fast=0;
//=====================================//
//...Choice of gauge + 3-body f-states...//
//=====================================//
  ForceUG=0;              /* to force Unitary Gauge assign 1 */
  VZdecay=0; VWdecay=0;   /* WW* and ZZ* channels are off */
```

Which modules are needed in this program

- Model in Feynman gauge
- Exclude 3/4-body f-states

A WIMPy benchmark

```
//=====================================//
//...Change external parameter values...//
//=====================================//
  assignValW("mss",   10.);
  assignValW("lamssh", 0.16);
```

Change DM mass and Higgs portal coupling

```
//=========================================================//
//...Sort odd particle spectrum and identify DM candidate(s)...//
//=========================================================//
  err=sortOddParticles(cdmName);
  qNumbers(CDM[1], &spin2, &charge3, &cdim);
```

Compute particle spectrum and identify dark matter candidate(s)

NB: sortOddParticles(cdmName) should be executed *every* time you modify parameter values

Thermal freeze-out

```
//=============================================//
//...Freeze-out scenario dark matter abundance...//
//=============================================//
#ifdef OMEGA
{ int fast=0;
  double Beps=1.E-4, cut=0.01;
  double Xf;
  double omegaFO;
  int i,err;

  printf("\n=============================================================================\n");
  printf("\n==== Calculation of relic density in thermal freeze-out scenario =====\n");
  printf("\n=============================================================================\n");
  printf("\n");

//...Compute the relic
  omegaFO=darkOmega(&Xf,fast,Beps,&err);

  printf("\n============================\n");
  printf("\n==== Freeze-out results =====\n");
  printf("\n============================\n");
  printf("\n");
  printf("Omega freeze-out=%.2e\n", omegaFO);
  printf("\n=============================================================================\n");

//...Print leading channels
  printChannels(Xf,cut,Beps,1,stdout);

  printf("\n=============================================================================\n");
}
#endif
```

Compute the thermal freeze-out DM abundance

Print out leading channels

Direct detection

Let’s calculate the spin-(in-)dependent scattering cross-section of protons/neutrons

```
//=========================================//
//...WIMP-nucleon scattering cross-sections...//
//=========================================//
#ifdef CDM_NUCLEON
{ double pA0[2],pA5[2],nA0[2],nA5[2];
  double Nmass=0.939; /*nucleon mass*/
  double SCcoeff;
  double csSIp1,csSIn1,csSDp1,csSDn1, csSIp1_,csSIn1_,csSDp1_,csSDn1_;

//...Compute the amplitudes for proton/neutron spin-independent/spin-dependent scattering
  nucleonAmplitudes(CDM[1], pA0,pA5,nA0,nA5);

//...Compute the corresponding cross-sections
  SCcoeff=4/M_PI*3.8937966E8*pow(Nmass*McdmN[1]/(Nmass+ McdmN[1]),2.);
  csSIp1=  SCcoeff*pA0[0]*pA0[0];
  csSDp1=3*SCcoeff*pA5[0]*pA5[0];
  csSIn1=  SCcoeff*nA0[0]*nA0[0];
  csSDn1=3*SCcoeff*nA5[0]*nA5[0];

  printf("\n==== Calculation of %s-nucleon cross sections [pb]: ====\n",CDM[1]);
  printf(" proton  SI %.3E SD %.3E \n", csSIp1,csSDp1);
  printf(" neutron SI %.3E SD %.3E \n", csSIn1,csSDn1);
  printf("\n============================================================\n");
}
#endif
```

Compute p,n amplitudes

Compute p,n cross-sections

A FIMPy benchmark

Let’s now redefine our coupling constant to something more appropriate for freeze-in

```
//=====================================================//
//...Redefine coupling, more suitable for freeze-in...//
//=====================================================//
  assignValW("lamssh", 1.995e-12);

  err=sortOddParticles(cdmName);
  qNumbers(CDM[1], &spin2, &charge3, &cdim);
```

Replace Higgs portal coupling

Freeze-in

```
//=============================================//
//...Freeze-in scenario dark matter abundance...//
//=============================================//
#ifdef FREEZEIN
{
  double TR=1E5;
  double omegaFI, omegaFIdec;

//...Add the lightest odd particle to the list of feeble particles
  toFeebleList(CDM[1]);

  printf("\n=================================================================\n");
  printf("\n==== Calculation of relic density in freeze-in scenario =====\n");
  printf("\n=================================================================\n");
  printf("\n");


//...Full calculation
  omegaFI=darkOmegaFi(TR,CDM[1],&err);

//...Calculation based on Higgs decays only
  omegaFIdec = darkOmegaFiDecay(TR, "H", "~ss");

  printf("\n============================\n");
  printf("\n==== Freeze-in results =====\n");
  printf("\n============================\n");
  printf("\n");
  printf("omega freeze-in=%.3E\n", omegaFI);
  printf("omega freeze-in from Higgs decays=%.3E\n", omegaFIdec);
  printf("\n=================================================================\n");

//...Print leading channels
  printChannelsFi(0,0,stdout);//...Full calculation


//...Always a good idea to empty the list of feeble particles after the computation
  toFeebleList(NULL);
}
#endif
```

Feebly interacting particles must be declared explicitly

- Full freeze-in calculation
- Calculation based on Higgs decay only

Print out leading channels

Empty the list of feeble particles

Beyond freeze-in

As a final example, let's check that solving a "freeze-in with backreactions" Boltzmann equation with zero initial abundances allows us to

i) Recover the two limits that we just saw

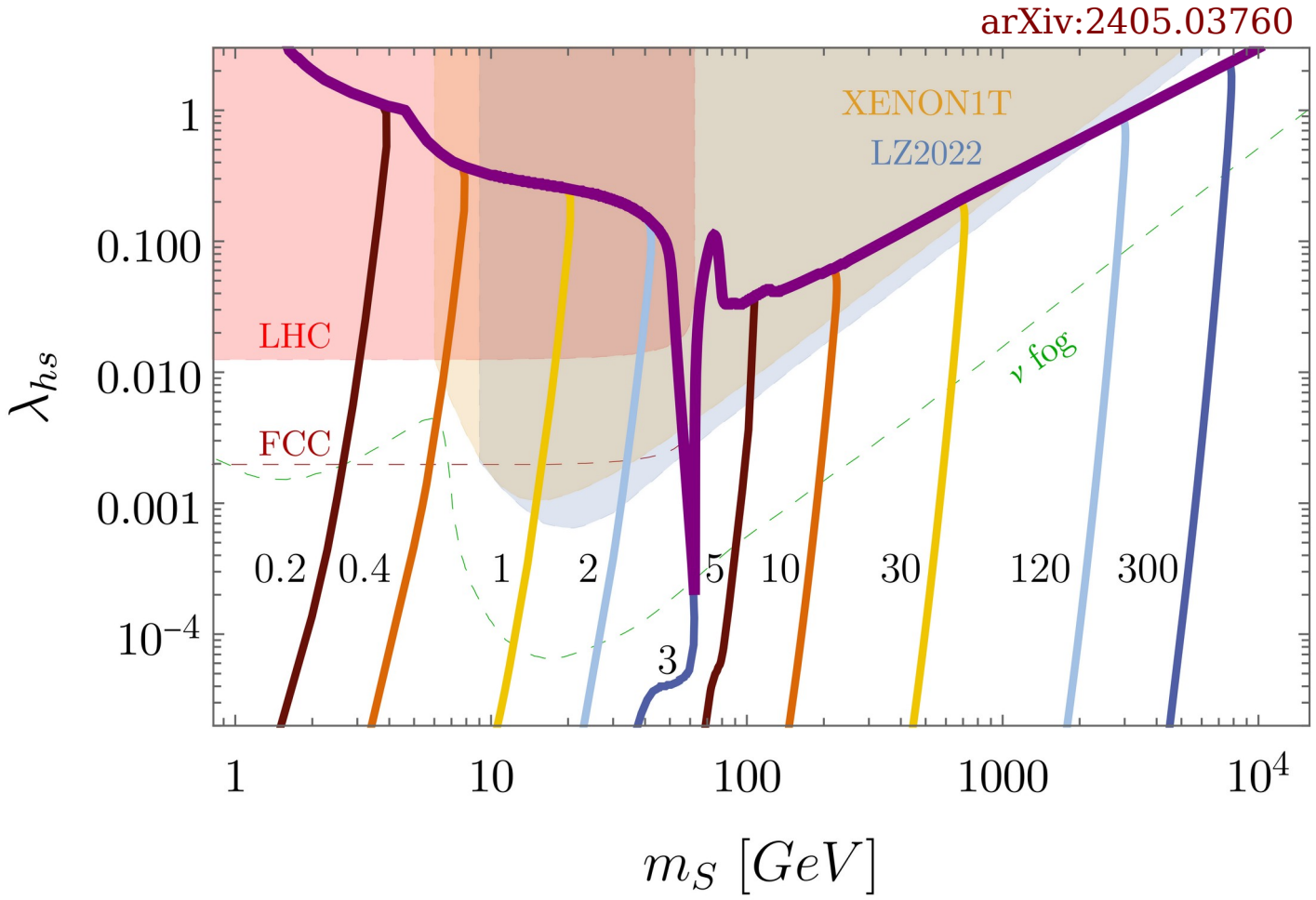
ii) Move beyond them

Please uncomment the last part of mymain.c

(well, except the comments!)

What did we just observe ?

Freeze-in with a low reheating temperature



Backreactions imply a smooth passage from freeze-in to freeze-out.

Thank you!