PROCEEDINGS OF SCIENCE

PoS(CompTools2021)037

An introduction to micrOMEGAs

Alexander Pukhov[a,*]

[a] *Skobeltsyn Institute of Nuclear Physics, Moscow, Russia*

E-mail: alexander.pukhov@gmail.com

MicroMEGAs is a code for the calculation of dark matter observables within generic extensions of the Standard Model. The code computes the relic density of a stable massive particle, the rates for direct and indirect detection of dark matter as well as the cross-sections at colliders and the decay widths of new particles. Here we introduce the main features of micrOMEGAs for the new user.

Computational Tools for High Energy Physics and Cosmology (CompTools2021)
22-26 November 2021
Institut de Physique des 2 Infinis (IP2I), Lyon, France

[*] Speaker

1. Introduction

Various observations indicate that dark matter (DM) forms the dominant matter component of the Universe and its relic density has been determined very precisely. Yet the exact nature of DM remains a mystery. Many particle physics models beyond the standard model (BSM) have proposed a DM candidate. Exploring the parameter space of these models is a challenging task. micrOMEGAs has been developed with the aim of computing automatically and precisely DM observables. These include the relic density; the fluxes of photons, antiprotons, and positrons for indirect DM searches; the cross sections and the nuclei recoil energy distribution for DM interactions with nuclei; the neutrino and the corresponding muon fluxes from DM particles captured by the Sun; the collider cross sections and partial decay widths of particles within a BSM that provides a possible WIMP (weakly Interacting Massive Particle) DM candidate. Moreover starting with version 4, micrOMEGAs can treat the case of two DM candidates and starting with version 5 the code also computes observables for feebly interacting DM [1]. micrOMEGAs is based on CalcHEP [2],

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

a code which generates automatically matrix elements and computes cross sections and particle decay widths in any particle physics model. micrOMEGAs has a modular structure, moreover it contains or imports external codes when required, for example, codes to compute the particle spectra in supersymmetric models or codes to constrain the model through its collider signatures. The operating system is Linux or Darwin (Mac) and the code is written in C.

2. Downloading of micrOMEGAs

The code can be downloaded from

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

The name of the file should be

```
micromegas_5.2.10.tgz
```

After unpacking the file, you will get the directory

```
micromegas_5.2.10
```

which occupies about 180 Mb of disk space. More disk space is required after compilation of specific models and generation of matrix elements.

3. File structure of micrOMEGAs.

The main directory of micrOMEGAs contains the following sub-directories

`CalcHEP_src/`	`generator of matrix elements`
`sources/`	`micrOMEGAs code`
`man/`	`description of micrOMEGAs routines in manual_5.2.pdf`
`Makefile/`	`to compile the kernel of the package`
`include/`	`include files for micrOMEGAs routines`
`lib/`	`for generated libraries`
`Packages/`	`for external codes`

as well as a directory for each DM model, namely

`MSSM/`	`Minimal Supersymmetric Standard Model`
`NMSSM/`	`Next-to-Minimal Susy Model`
`CPVMSSM/`	`MSSM with complex parameters`
`IDM/`	`Inert doublet model`
`LHM/`	`Little Higgs Model`
`SingletDM/`	`Singlet scalar model with Z2 symmetry`
`Z3IDM/`	`Inert doublet and singlet with Z3 symmetry`
`Z4IDSM/`	`Inert doublet and singlet with Z4 symmetry`
`Z5M/`	`Two singlets model with Z5 symmetry`
`ZpPortal/`	`Z' portal model`
`LLL_scalar/`	`Model with singlet charged lepton and real scalar DM`

Each model directory has the same structure, it contains

`main.c/`	`sample main program for the given model`
`Makefile/`	`to compile  main.c or C and C++ user's codes`
`lib/`	`source codes of specific model routines`
`work/`	`CalcHEP working directory intended for matrix element generation`

In each model, the work directory contains the subdirectories

`models/`	`files which describe the model in CALCHEP format:`
	`vars1.mdl func1.mdl prtcls1.mdl lgrng1.mdl`

`so_generated/`	`to store the automatically generated matrix elements`

The command `./calchep` in the `work` directory launches an interactive CalcHEP session that allows to test the model.

The directory `Packages` contains several external codes. In particular some codes intended to calculate loop-improved spectra for particles in MSSM-like models, namely `SuSpect` for the MSSM [3], `NMSSMTools` for the NMSSM [4], and `CpsuperH` for the MSSM with complex parameters[5]. The code `LoopTools`[33] used for `DM,DM` $\to \gamma\gamma$ is also included as well as `Lilith` [6, 7], which imposes experimental constraints on the properties of the Higgs boson. In addition other codes will be downloaded automatically when the user calls the corresponding functions. These include:

`HiggsBounds` – experimental constraints from searches of scalar particles [8]
`HiggsSignals` – experimental constraints on the properties of the Higgs [9]
`SModelS` – LHC constraints from searches for new particles [10]
`SPheno` [11] and `SoftSUSY` [12]– calculation of particle spectra in the MSSM
`SuperIso` [13]– flavour physics constraints.

The `main.c, main.cpp` files in micrOMEGAs model directories consist of several independent blocks enclosed within the lines

```
   #ifdef XXXXX
     ............
   #endif
```

At the top of the main.c file the user can switch on/off any block via the corresponding #define instruction

```
#define MASSES_INFO          // Display information about mass spectrum
#define CONSTRAINTS          // Model-dependent constraints
#define HIGGSBOUNDS          // Constraints from Higgs searches
#define HIGGSSIGNALS         // Constraints from Higgs properties
#define LILITH               // Constraints from Higgs properties
#define OMEGA                // Calculate relic density
#define INDIRECT_DETECTION   // Spectra of gamma/e+/anti-p/neutrino from
                             // DM annihilation
#define RESET_FORMFACTORS    // Redefinition of Nucleus Form Factors
#define CDM_NUCLEON          // Calculate amplitudes and cross-sections for
                             // DM-nucleon collision
#define CDM_NUCLEUS          // Calculate number of events and their energy
                             // distribution for DM nucleus collision
#define NEUTRINO             // Neutrino telescope
#define DECAYS               // Widths and  branching ratios
#define CROSS_SECTIONS       // Calculate cross sections for colliders
#define SHOWPLOTS            // Display  plots on the screen
#define CLEAN                // Remove intermediate files
```

The main.c files from all model directories are similar and call the same micrOMEGAs routines.

4. Compilation of micrOMEGAs

To compile the code and use all the functions of micrOMEGAs it is desirable to have gfortran[1] and X11 development files /usr/include/X11/*.h. Fortran is required for some models distributed with micrOMEGAs. X11 is required for both the Graphic User Interface in CalcHEP, see Section 8, and for the various plotting facilities of micrOMEGAs. One can ignore this requirement or update X11, for example, with

[1] Note that on MAC, gfortran must be reinstalled after updating the system.

```
Xquartz (https://www.xquartz.org) on Mac
libX11-dev for Ubuntu/Debian
xorg-x11-devel for SUSE
```

Note that the external code SModelS requires python 3.0 or a more recent version.

To compile CalcHEP and the main micrOMEGAs routines one has to call

```
make
```

in the micrOMEGAs directory. The automatically detected compiler flags are stored in the file

```
CalcHEP_src/FlagsForSh
```

The user can change them and call `make` again. Any model directory contains a file `main.c` which can be modified by the user. The command `make main=`*my_main.c[pp]* generates the executable file

```
./my_main
```

`make` called without parameters works like "`make main=main.c`" and generates executable `main` which requires one argument, the name of the file which contains the input parameters of the model. For instance,

```
./main data1.par
```

If micrOMEGAs needs a matrix element for some process, it calls `CalcHEP` to generate this matrix element. The code for this matrix element is compiled, and stored as a shared library in the directory

```
NAME_OF_MODEL/work/so_generated
```

The user sees a message on the screen

```
PROCESS: <name of process>
```

or `VERTEX : < name of vertex>`

The shared library is loaded dynamically in run time. Each shared library is generated only once. When one of the model files is changed, the shared library containing the matrix elements is recompiled automatically.

5. Particle physics models in micrOMEGAs

The code micrOMEGAs assumes the existence of a discrete symmetry responsible for the stability of Dark Matter. Typically it is a Z_2 symmetry such as R-parity in the MSSM. This symmetry divides all particles in two classes, *odd* and *even*. The lightest *odd* particle is stable and will be treated as Dark Matter. For micrOMEGAs the *odd* particles are those whose name starts with tilde "~". micrOMEGAs can also work with models with two DM, the particles of the two dark sectors are those whose names start with "~" and "~~". The first DM will be the lightest "~" particle while the second DM will be the lightest "~~" particle. These can be found in models with discrete symmetries such as $Z_2 \times Z_2$ or Z_4.

5.1 Example: Inert Doublet Model

To examine the files that describe a dark matter model let us consider the inert doublet model. This model contains two $SU(2) \times SU(1)$ doublet [14]

$$H_1 = \begin{pmatrix} 0 \\ v + h/\sqrt{2} \end{pmatrix} \qquad H_2 = \begin{pmatrix} H^+ \\ (X + iH_3)/\sqrt{2} \end{pmatrix} \tag{1}$$

where H_1 corresponds to the SM doublet. The Lagrangian contains only even powers of the H_2 doublet

$$\mathcal{L} = \mathcal{L}_{SM} + D^{\mu} H_2^* D_{\mu} H_2 - \mu^2 H_2^2 - \lambda_2 H_2^4 - \lambda_3 H_1^2 H_2^2 - \lambda_4 |H_1^* H_2|^2 - \lambda_5 Re[(H_1^* H_2)^2] \tag{2}$$

Because of the symmetry $H_2 \to -H_2$ the lightest among $H^{\pm}$, X, H_3 is stable. The fundamental parameters μ, λ_3, λ_5 can be expressed in terms of masses, leaving only two new couplings, λ_2 and $\lambda_L = \lambda_3 + \lambda_4 + \lambda_5$, more details can be found in Ref. [14]. The model files are written in the `CalcHEP` format, each model contains four files stored in the IDM/work/models directory.

The first file, `vars1.mdl`, contains the free parameters of the model, below is an extract of the parameter file of the IDM,

```
Inert Doublet Model
Variables
Name    | Value    |>             Comment             <|
EE      |0.31333   |Electromagnetic coupling constant
SW      |0.474     |sin of the Weinberg angle
MZ      |91.187    |Mass of Z
MHX     |111       |Mass of Inert Doublet Higgs
MH3     |222       |Mass of CP-odd Higgs
MHC     |333       |Mass of charged Higgs
LaL     |0.01      |Coupling in Inert Sector
```

The second file `func1.mdl` contains the constrained parameters of the model, for example,

```
Inert Doublet
Constraints
Name |> Expression
CW   |sqrt(1-SW^2)
MW   |MZ*CW
Mb   |MbEff(Q)
Mc   |McEff(Q)
mu2  |MHX^2-laL*(2*MW/EE*SW)^2
la3  |2*(MHC^2-mu2)/(2*MW/EE*SW)^2
la5  |(MHX^2-MH3^2)/(2*MW/EE*SW)^2
```

The third file `prtcls1.mdl` contains the list of particles of the model, note that the names of all particles in the odd sector start with tilde "~".

```
Full   Name     | P | aP|  number  |spin2|mass|width|color|aux|> LaTeX(A)
photon          |A  |A  |22        |2    |0   |0    |1    |G  |A
Z boson         |Z  |Z  |23        |2    |MZ  |!wZ  |1    |G  |Z
gluon           |G  |G  |21        |2    |0   |0    |8    |G  |G
W boson         |W+ |W- |24        |2    |MW  |!wW  |1    |G  |W^+
neutrino        |n1 |N1 |12        |1    |0   |0    |1    |L  |\nu^e
electron        |e1 |E1 |11        |1    |0   |0    |1    |   |e
mu-neutrino     |n2 |N2 |14        |1    |0   |0    |1    |L  |\nu^\mu
muon            |e2 |E2 |13        |1    |Mm  |0    |1    |   |\mu
tau-neutrino    |n3 |N3 |16        |1    |0   |0    |1    |L  |\nu^\tau
tau-lepton      |e3 |E3 |15        |1    |Mt  |0    |1    |   |\tau
u-quark         |u  |U  |2         |1    |0   |0    |3    |   |u
d-quark         |d  |D  |1         |1    |0   |0    |3    |   |d
c-quark         |c  |C  |4         |1    |Mc  |0    |3    |   |c
s-quark         |s  |S  |3         |1    |Ms  |0    |3    |   |s
t-quark         |t  |T  |6         |1    |Mtop|wtop |3    |   |t
b-quark         |b  |B  |5         |1    |Mb  |0    |3    |   |b
Higgs           |h  |h  |25        |0    |Mh  |!wh  |1    |   |h
odd Higgs       |~H3|~H3|36        |0    |MH3 |!wH3 |1    |   |(H3)
Charged Higgs   |~H+|~H-|37        |0    |MHC |!wHC |1    |   |(H+)
second Higgs    |~X |~X |35        |0    |MHX |!wHX |1    |   |(X)
```

Finally the vertices of the model are listed in the `lgrng1.mdl` file, some vertices are listed below

```
Inert Doublet
Lagrangian
P1 |P2 |P3 |P4 |> Factor         <|> dLagrangian/ dA(p1) dA(p2)dA(p3)
A  |W+ |W- |   |-EE               |m3.p2*m1.m2-m1.p2*m2.m3- ......
A  |~H+|~H-|   |EE                |m1.p3-m1.p2
B  |b  |A  |   |EE/3              |G(m3)
B  |b  |G  |   |GG                |G(m3)
B  |b  |Z  |   |-EE/(12*CW*SW)    |4*SW^2*G(m3)-3*G(m3)*(1-G5)
B  |b  |h  |   |-EE*Mb/(2*MW*SW)  |1
B  |t  |W- |   |-EE*Sqrt2/(4*SW)  |G(m3)*(1-G5)
W+ |W- |~X |~X |EE^2/(2*SW^2)     |m1.m2
h  |~X |~X |   |-2*MW*SW/EE       |la3+la4+la5
Z  |Z  |~X |~X |EE^2/(2*CW2*SW^2) |m1.m2
.................
```

Here `pi` is the momentum or particle `Pi`, and `m` denotes a Lorentz index, for example, in the $A^{\mu}W^{+\nu}W^{-\delta}$ vertex, m3.p2*m1.m2 stands for $p_2^{\delta}g^{\mu\nu}$.

6. Main functions of micrOMEGAs.

Here we describe some of the functions of micrOMEGAs that can be used to compute DM observables in any of the models provided with the code or in a user-defined model. Most of these functions can be found in the sample `main.c` files included in any of the model directories.

6.1 Changing free model parameters.

To compute DM observables for a single point in the parameter space of a given model it is convenient to define the free parameters in an input file. All parameters that are not defined explicitly will be assigned their default value given in `vars1.mdl`. The structure of the input file has to be

```
name  value [ # comment ]
```

For instance, in the IDM a sample input file has the form

```
laL 0.001  # coupling
MHX   600  # inert sector Higgs
Mh    125  # SM Higgs mass
la2  0.01  # coupling
MHC   604  # mass of charged Higgs
MH3   601  # mass of CP odd Higgs
```

The function to download the set of free parameters is `readVar(fileName)`. Any of the free parameters can be redefined using `assignValW(name,value)` which assigns a new value to the parameter `name`, for example, `assignValW(“MHX”,600).` After assigning new values to the free parameters one has to call

```
    sortOddParticles(outText)
```

in order to recalculate the constrained parameters of the model and to find the lightest odd particle[s] which will be the DM particle[s]. These are defined as `CDM1` [`CDM2`] with masses `Mcdm1 [Mcdm2]`. Morever the parameter `Mcdm` is defined as `min(Mcdm1,Mcdm2)`. When an error occurs in the calculation of the constrained parameter, this routine returns an error code and `outText` contains the name of the parameter which cannot be calculated.

The values of the constrained and free parameters can be obtained with `findValW(name)` and the masses of particles can be obtained with the function `pMass(name)`.

6.2 Calculation of the DM relic density

The solution of the DM relic density follows the standard methods proposed in [15] and described in [16]. The evolution equation for the DM number density reads,

$$\frac{dn}{dt} = - < v\sigma > (n^2 - n_{eq}^2) - 3Hn \tag{3}$$

where H is the Hubble parameter, n_{eq} is is the equilibrium DM number density and $\langle v\sigma \rangle$ is the thermally average DM annihilation cross-section. A simpler equation can be written in terms of the

DM abundance, $Y = n_{DM}/s$ where s is the entropy density. The law of entropy conservation

$$\frac{ds}{dt} = -3Hs \tag{4}$$

leads to a cancellation of the last term in Eq. 3. The relation between the entropy density s and the temperature reads

$$s(T) = \frac{2\pi^2}{45}T^3 h_{eff}(T) \tag{5}$$

where $h_{eff}(T)$ counts the effective number of SM degrees of freedom. This allows to rewrite the evolution equation for the abundance as a function of temperature using,

$$\frac{d}{dt} = -3Hs\frac{d}{ds} = -\frac{3Hs}{ds/dT}\frac{d}{dT} \tag{6}$$

we get,

$$\frac{dY}{dT} = \frac{< v\sigma >}{3Hs}\frac{ds}{dT}(Y^2 - Y_{eq}^2) \tag{7}$$

The relic density of dark matter is obtained simply after solving for the abundance today.

The main function to compute the relic density when the model contains only one DM is `darkOmega(&Xf,fast,Beps)`
This function uses the Runge-Kutta method to solve the evolution equation for the abundance and returns Ωh^2 as well as X_f =`Mcdm`$/T_f$, where T_f is the freeze-out temperature - the temperature of DM formation. It is defined by the condition

$$Y(T_f) = 2.5Y_{eq}(T_f) \tag{8}$$

The flag `fast =1` allows for fast calculation, here `micrOMEGAs` uses a Gauss n-point integration with separation of s-channel poles in matrix elements; `fast=0` uses Simpson adaptive integration, and `fast=-1` uses Simpson adaptive integration with separation of s-channel poles, see details in [17].

The Hubble parameter H and effective number of degrees of freedom h_{eff} can be accessed through the functions, `Hubble(T)` and `hEff(T)`. The equilibrium abundance and the solution for the DM abundance are represented by the functions `Yeq(T)` and `YF(T)` respectively.

We refer the reader to the manual [18] for the generalisation to the case of two-component Dark Matter.

6.3 Direct Detection

To predict the results of direct detection experiments in a given model one needs to calculate the recoil energy distribution and the cross-sections for DM - nuclei scattering. Since the velocities of DM particles in the halo of the Milky Way are of the order of the orbital velocities of stars 0.001c, elastic cross sections are computed in the $v = 0$ limit after writing an effective Lagrangian for *Spin Independent* (scalar) and *Spin Dependent* (spin flip) DM scattering amplitudes. Using Feynman rules one can compute the DM - quarks cross sections, then nucleon form factors are used to compute the DM - nucleon scattering cross section. Finally nuclei form factors are introduced to compute DM -nucleus cross sections.

The function that computes amplitudes for DM scattering on nucleons is

`nucleonAmplitudes(name_of_DM ,pA0,pA5,nA0,nA5)`

where the output `pA0,pA5,nA0,nA5` are two dimensional arrays which contain the amplitudes for DM and anti-DM scattering on nucleons

	spin independent	spin dependent
Proton	pA0	pA5
Neutron	nA0	nA5

The DM-nucleon cross sections are given by,

$$\begin{aligned} \sigma^{SI} &= C \cdot A^2 \\ \sigma^{SD} &= 3C \cdot A^2 \end{aligned} \tag{9}$$

where

$$C = \frac{4}{\pi}\left(\frac{M_N M_{dm}}{M_M + M_{dm}}\right)^2 \cdot 3.89 \cdot 10^8 \text{pb} \ , \tag{10}$$

M_N is the nucleon mass, M_{dm} the mass of the DM particle and A is the amplitude computed by `nucleonAmplitudes`. The latter depends on the form factors that describe the quark content in the nucleon. In the code, these form factors are global parameters that can be easily changed by the user, details can be found in [19].

The cross sections for elastic DM scattering on protons calculated with `nucleonAmplitudes` can be directly compared with the limits obtained by Direct Detection experiments assuming the standard DM density and velocity distribution in the Milky Way. Moreover these limits are valid for point-like DM-nucleon interactions and assuming identical cross-sections for protons and neutrons. These assumptions can be violated, for example, when interactions of DM with nucleons is mediated by low mass mediators [20], when there are more than one DM or for alternative DM velocity distribution. For these cases a recast of four Direct Detection experiments XENON [21, 22], DarkSide [23], PICO [24] and CRESST [25] performed within micrOMEGAs can be used. The function

`DD_pval(expCode, fv,&expName)`

returns 1-C.L. for the given DM scenario after taking into account the four experiments mentioned above. Here `expCode` is the sum of the binary codes for the experiments

```
XENON1T_2018,  DarkSide_2018, PICO_2019, CRSST_2019
```

The DM velocity distribution is defined by the function `fv`. `expName` is a return parameter which contains the name of the experiment which gives the strongest exclusion.

6.4 Indirect Detection

DM annihilation in the Galactic halo produces pairs of standard model particles that hadronize and decay into stable particles. The final states with $\gamma, e^+, \bar{p}, \nu$ are the subject of indirect searches. First, micrOMEGAs calculates the spectra of stable particles produces in DM collisions:

`calcSpectrum(key,Sg,Se,Sp,Sne,Snm,Snl,&err)`.

This function returns $v \cdot \sigma$, the cross section for DM annihilation in cm^3/sec units at $v = 10^{-3}c$,

and calculates the spectra of photons, positrons, antiprotons, and neutrinos (ν_e, ν_μ, ν_τ). The spectra are stored in the arrays `Sg, Se, Sp, Sne, Snm, Snl` respectively. These arrays have dimension NZ=250. Here CalcHEP is used to calculate the primary annihilation cross section as well as the decay products of non-SM particles, while PYTHIA 6.4 [26] is used to calculate the different spectra after hadronisation of the SM particles produced in DM pair annihilation. When the model contains several types of DM, a summation over all types of DM - DM/anti-DM reactions is performed. The `key` parameter is a binary code which has the following meaning:
1-takes into account W/Z polarization
2-includes gammas from 2->2+gamma
4-prints cross sections for different channels

An interpolation function, `SpectdNdE(E,Spect)` can be used for any of these arrays. Moreover, one can use the `displayPlot` routine presented in Section 7 to visualize and compare different spectra.

The number of DM annihilation depends on the square of the DM local density, `rhoDM`. The default value for this global parameter of micrOMEGAS is 0.3GeV/cm^3 [27] and can be changed by the user.

The value of $v \cdot \sigma$ for a given DM annihilation final state can be directly compared with the limits derived experimentally, for example, by FermiLAT in the photon channel. For a more complete estimation of the signal, one needs to compute the fluxes of photons, for this one has to take into account the DM distribution in the Milky Way and integrate the photon signal over the line of sight. This can be done with the function

`gammaFluxTab(fi,dfi,sigmav,Sg,Sobs)`

which transforms the photon spectrum from DM pair annihilation `Sg` to an observable tabulated photon flux `Sobs`. Here `fi` is the angle between the line of sight and the center of the galaxy; `dfi` is half the cone angle which characterizes the detector resolution (the solid angle is $2\pi(1-\cos(\texttt{dfi}))$; and `sigmav` is the annihilation cross section calculated by `calcSpectrum`. The interpolation function `SpectdNdE(E,Sobs)` returns the resulting photon flux in [1/(GeV cm^2 s)] units.

The DM distribution profile is taken to be the Zhao profile, [28]

$$F_{halo}(r) = \left(\frac{R_\odot}{r}\right)^\gamma \left(\frac{r_c^\alpha + R_\odot^\alpha}{r_c^\alpha + r^\alpha}\right)^{\frac{\beta-\gamma}{\alpha}} \tag{11}$$

with by default $\alpha = 1, \beta = 3, \gamma = 1, rc = 20[kpc]$. The parameters of the Zhao profile can be reset using `setProfileZhao(`α,β,γ,rc`)`

6.4.1 Antiproton and positron fluxes

The positron and antiproton fluxes are computed based on a semi-analytical model which describes the propagation of charged particles in the Galaxy [29]. The free parameters of this model are global parameters of micrOMEGAs,

K_dif	0.0112	kpc^2/Myr	Normalized diffusion coefficient
L_dif	4	kpc	Vertical size of the diffusive zone
Delta_dif	0.7		Slope of the diffusion coefficient
Tau_dif	10^{16}	s	Electron energy loss time
Vc_dif	0	km/s	Convective Galactic wind

The positron and antiproton fluxes are calculated respectively with

```
posiFluxTab(Emin,sigmav, Se, Sobs)
pbarFluxTab(Emin,sigmav, Sp, Sobs)
```

where `Sobs` is an array which contains the resulting spectrum. This array can be interpolated with `SpectdNdE(E,Sobs)` which returns the flux in [1/(GeV cm^2 s sr)] units.

7. Plots in micrOMEGAs

The routine that allows to display plots on the screen is

```
displayPlot(title,xName,xMin,xMax,lScale,N, ...)
```

It can display several curves/histograms on one plot. Here `xName` is the name of the x variable; `xMin,xMax` are the lower and upper limits of x; a logarithmic scale for the x axis is used if `lScale`$\neq$ 0; `N` is the number of curves/histograms to display. After the parameter N, `displayPlot` expects 4N parameters, where each tetrad contains

1) text label of curve;

2) dimension of array or 0 for functions;

3) array of data or a function to display

4) array of error, or an address of the function arguments, or NULL if the function does not depend implicitly on some parameters.

Figures 1,2 show examples of plots generated by micrOMEGAs, the corresponding captions contain the commands which generate the plots. Fig.1 illustrates how to save the data of the plot in Root, PAW, GnuPlot or Python. The information saved consists of two files, the code written in the language requested and the data table. The user can of course improve the code to adapt it for his needs. Clicking on the plot with the mouse one can see the value of the functions for a chosen value of x . Fig.2 shows the case with an argument corresponding to the array where the spectrum is stored.

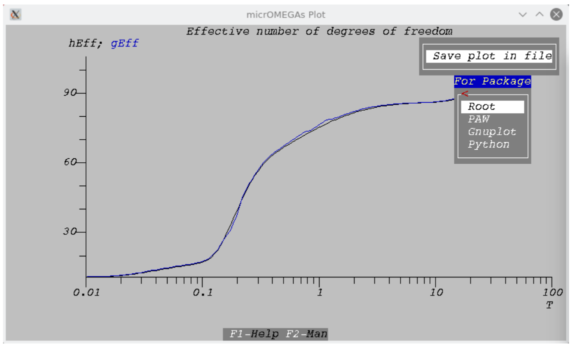


Figure 1: displayPlot("Effective number of degrees of freedom","T",0.01,100,1,2, "hEff",0,hEff,NULL, "gEff",0,gEff,NULL);

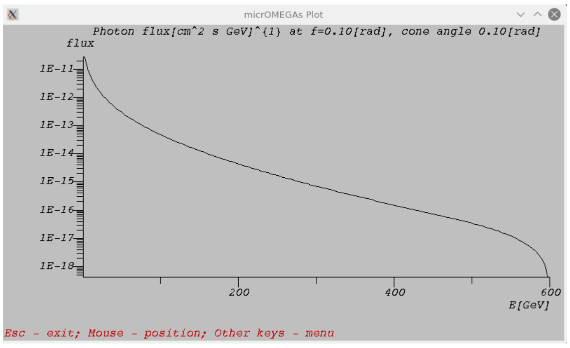


Figure 2: displayPlot(txt,"E[GeV]",Emin,Mcdm,0,1, "flux",0,SpectdNdE,Sobs);

8. Implementation of new models

In addition to the several models distributed with the code, the user can easily define a new dark matter model. For this the user must launch the command

```
./newProject MODEL
```

from the main micrOMEGAs directory. This will create the directory MODEL, which will contain all the files needed to run micrOMEGAs with the exception of the new model files. The latter should be written in the CalcHEP format and included in the sub-directory `MODEL/work/models`. The files needed are `vars1.mdl, func1.mdl, prtcls1.mdl, lgrng1.mdl`.

For example, the following steps will allow to create a new model and copy the model files from an existing model:

```
./newProject IDMcopy
  cp IDM/work/models/*1.mdl   IDMcopy/work/models
  cp IDM/*.dat   IDMcopy/.
```

The model files for new models can be created by using one of the following programs
`LanHEP`[30], `FeynRules` [31] or `Sarah` [32]
LanHEP is included in micrOMEGAs and most model directories contain a lanhep subdirectory with the necessary source files and with the Makefile required to call LanHEP. One can construct a new model following these examples. More complete information can be found in the LanHEP manual

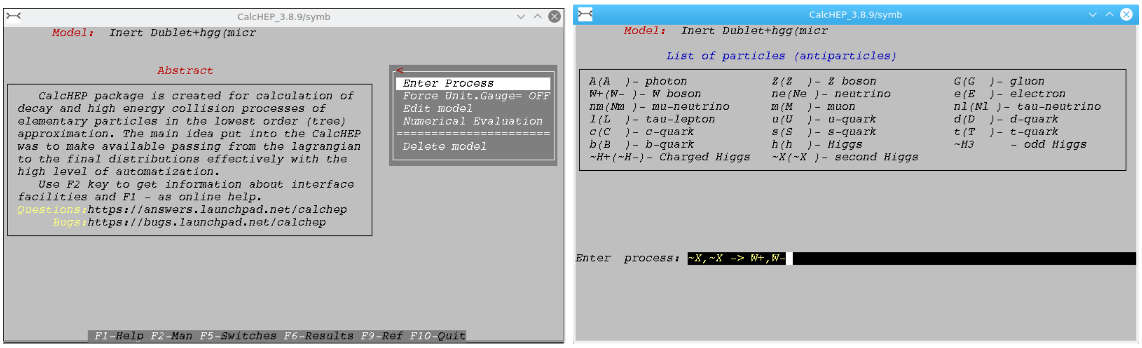
```
micromegas_5.2.10/Packages/LanHEP/manual/man31.pdf
```

The command `make` launched from the MODEL directory checks the model and stops with an error code when the model files do not correspond to CalcHEP requirements. The user can go to the `work` directory, launch `./calchep` and use the menu line

```
CHECK MODEL
```

to improve the files using the CalcHEP menu system, see Fig. 3.

A CalcHEP GUI session is a powerful tool for a better understanding of the DM model and of the processes relevant for DM formation. For example, one can examine the generated Feynman diagrams before doing the symbolic calculation of the squared matrix element for a given process, see Fig. 4. After calling the c-compiler, the user can read parameters from a data file in the micrOMEGAs directory, Fig. 5 (left). Note that in order to load parameters stored in a file from the



```
      Model:  Inert Dublet+hgg(micr

                         Abstract
   CalcHEP package is created for calculation of
decay and high energy collision processes of
elementary particles in the lowest order (tree)
approximation. The main idea put into the CalcHEP
was to make available passing from the lagrangian
to the final distributions effectively with the
high level of automatization.
   Use F2 key to get information about interface
facilities and F1 - as online help.
Questions:https://answers.launchpad.net/calchep
     Bugs:https://bugs.launchpad.net/calchep
```

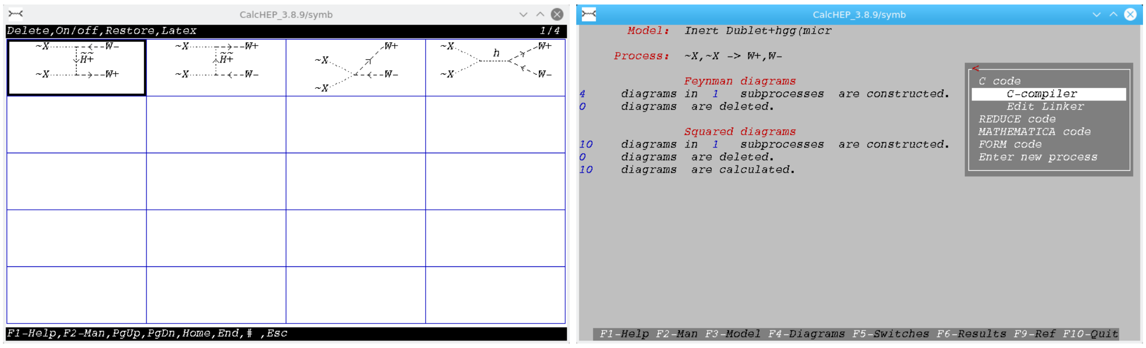
```
      Model:  Inert Dublet+hgg(micr

           List of particles (antiparticles)
 A(A   )- photon           Z(Z   )- Z boson          G(G   )- gluon
 W+(W- )- W boson          ne(Ne )- neutrino         e(E   )- electron
 nm(Nm )- mu-neutrino      m(M   )- muon             nl(Nl )- tau-neutrino
 l(L   )- tau-lepton       u(U   )- u-quark          d(D   )- d-quark
 c(C   )- c-quark          s(S   )- s-quark          t(T   )- t-quark
 b(B   )- b-quark          h(h   )- Higgs            ~H3      - odd Higgs
 ~H+(~H-)- Charged Higgs   ~X(~X )- second Higgs

Enter  process: ~X,~X -> W+,W-
```

Figure 3: Left: Menu which allows to view and modify the model files, and to check constraints, particle masses, widths, and decay branching ratios. Right: Entering a process in CalcHEP.

model directory one one has to use the path

```
../../data1.par
```

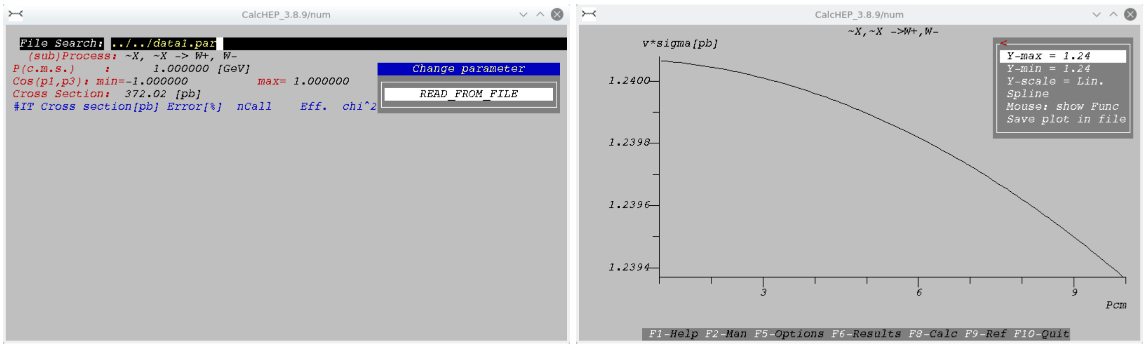
since the numerical session is launched from the directory `work/results`.



```
      Model:  Inert Dublet+hgg(micr

    Process:  ~X,~X -> W+,W-

                 Feynman diagrams
4      diagrams in  1   subprocesses  are constructed.
0      diagrams  are deleted.

                 Squared diagrams
10     diagrams in  1   subprocesses  are constructed.
0      diagrams  are deleted.
10     diagrams  are calculated.
```

Figure 4: Left: Generated Feynman diagrams. Right: Call of C-compiler.

One can also compute $v \cdot \sigma$, Fig. 5 (right), which is a simplified version of the thermally averaged cross-section $< v\sigma >$ in Eq.7. Note that in order to reproduce $\Omega h^2 = 0.12$, the typical cross-section should be around $v \cdot \sigma \approx 1\text{pb} \cdot \text{c}$. However this can be modified when co-annihilation processes are important.



```
 File Search: ../../data1.par
  (sub)Process: ~X, ~X -> W+, W-
P(c.m.s.)    :      1.000000 [GeV]
Cos(p1,p3): min=-1.000000         max= 1.000000
Cross Section:  372.02 [pb]
#IT Cross section[pb] Error[%]  nCall    Eff.  chi^2
```

Figure 5: Left: Reading parameters from file disposed in a micrOMEGAs directory. Right: Calculation of $v \cdot \sigma$ contributing to the relic density.

9. Conclusion

A complete description of the code micrOMEGAs can be found in the manual contained in the `man` directory and in the various publications related to the code [16, 17, 19, 27, 34]. Moreover a tutorial which in particular explains how to create a new model can be found in [35].

10. Acknowledgements

I am grateful for the organizers of the conference for the invitation to make this report. The code micrOMEGAs and the interfaces to other codes was developed in collaboration with D. Barducci, G. Bélanger, A. Belyaev, J. Bernon, F. Boudjema, P. Brun, J. Da Silva, A. Goudelis, C. Hugonie, S. Kraml, U. Laa, A. Mjallal, S. Rosier-Lees, P. Salati, A. Semenov and B. Zaldívar. The work was supported in part by a grant RFBR-20-52-15005 and by a grant AAP-USMB. It was also supported by the Interdisciplinary Scientific and Educational School of Moscow University Fundamental and Applied Space Research and was done within the framework of the scientific program of the National Center for Physics and Mathematics, the project "Particle Physics and Cosmology".

References

[1] G. Bélanger, F. Boudjema, A. Goudelis, A. Pukhov and B. Zaldivar, "micrOMEGAs5.0 : Freeze-in," *Comput. Phys. Commun.* **231** (2018) 173-186, [arXiv:1801.03509 [hep-ph]].

[2] A. Belyaev, N. D. Christensen and A. Pukhov, "CalcHEP 3.4 for collider physics within and beyond the Standard Model," *Comput. Phys. Commun.* **184** (2013) 1729-1769, [arXiv:1207.6082 [hep-ph]].

[3] A. Djouadi, J.-L. Kneur, and G. Moultaka, "SuSpect: A Fortran code for the supersymmetric and Higgs particle spectrum in the MSSM," *Comput. Phys. Commun.* **176** (2007) 426–455, `arXiv:hep-ph/0211331`.

[4] U. Ellwanger and C. Hugonie, "NMSPEC: A Fortran code for the sparticle and Higgs masses in the NMSSM with GUT scale boundary conditions," *Comput. Phys. Commun.* **177** (2007) 399–407, `arXiv:hep-ph/0612134 [hep-ph]`.

[5] J. S. Lee, A. Pilaftsis, M. Carena, S. Y. Choi, M. Drees, J. R. Ellis, and C. E. M. Wagner, "CPsuperH: A Computational tool for Higgs phenomenology in the minimal supersymmetric standard model with explicit CP violation," *Comput. Phys. Commun.* **156** (2004) 283–317, `arXiv:hep-ph/0307377 [hep-ph]`.

[6] J. Bernon and B. Dumont, "Lilith: a tool for constraining new physics from Higgs measurements," *Eur. Phys. J.* **C75** no. 9, (2015) 440, `arXiv:1502.04138 [hep-ph]`.

[7] M. Bertrand, S. Kraml, T. Q. Loc, D. T. Nhung, and L. D. Ninh, "Constraining new physics from Higgs measurements with Lilith-2," in *Tools for High Energy Physics and Cosmology*. 12, 2020. `arXiv:2012.11408 [hep-ph]`.

[8] P. Bechtle, O. Brein, S. Heinemeyer, O. Stål, T. Stefaniak, *et al.*, “HiggsBounds – 4: Improved Tests of Extended Higgs Sectors against Exclusion Bounds from LEP, the Tevatron and the LHC,” *Eur. Phys. J.* **C74** (2014) 2693, arXiv:1311.0055 [hep-ph].

[9] P. Bechtle, S. Heinemeyer, T. Klingl, T. Stefaniak, G. Weiglein and J. Wittbrodt, “HiggsSignals-2: Probing new physics with precision Higgs measurements in the LHC 13 TeV era,” *Eur. Phys. J.* **C81** (2021) no.2, 145, arXiv:2012.09197 [hep-ph]].

[10] F. Ambrogi, J. Dutta, J. Heisig, S. Kraml, S. Kulkarni, U. Laa, A. Lessa, P. Neuhuber, H. Reyes-González and W. Waltenberger, *et al.* “SModelS v1.2: long-lived particles, combination of signal regions, and other novelties,” *Comput. Phys. Commun.* **251** (2020) 106848, arXiv:1811.10624 [hep-ph].

[11] W. Porod and F. Staub, “SPheno 3.1: Extensions including flavour, CP-phases and models beyond the MSSM,” *Comput. Phys. Commun.* **183** (2012) 2458–2469, arXiv:1104.1573 [hep-ph].

[12] B. C. Allanach, “SOFTSUSY: a program for calculating supersymmetric spectra,” *Comput. Phys. Commun.* **143** (2002) 305–331, arXiv:hep-ph/0104145 [hep-ph].

[13] F. Mahmoudi, “SuperIso v2.3: A Program for calculating flavor physics observables in Supersymmetry,” *Comput. Phys. Commun.* **180** (2009) 1579–1613, arXiv:0808.3144 [hep-ph].

[14] M. Gustafsson, “The Inert Doublet Model and Its Phenomenology,” *PoS* **CHARGED2010** (2010) 030, arXiv:1106.1719 [hep-ph].

[15] P. Gondolo and G. Gelmini, “Cosmic abundances of stable particles: Improved analysis,” *Nucl. Phys. B* **360** (1991) 145–179.

[16] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, “MicrOMEGAs: A Program for calculating the relic density in the MSSM,” *Comput. Phys. Commun.* **149** (2002) 103–120, arXiv:hep-ph/0112278 [hep-ph].

[17] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, “MicrOMEGAs 2.0: A Program to calculate the relic density of dark matter in a generic model,” *Comput. Phys. Commun.* **176** (2007) 367–382, arXiv:hep-ph/0607059 [hep-ph].

[18] G. Bélanger, F. Boudjema, A. Pukhov, and A. Semenov, “micrOMEGAs4.1: two dark matter candidates,” *Comput. Phys. Commun.* **192** (2015) 322–329, arXiv:1407.6129 [hep-ph].

[19] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, “Dark matter direct detection rate in a generic model with micrOMEGAs 2.2,” *Comput. Phys. Commun.* **180** (2009) 747–767, arXiv:0803.2360 [hep-ph].

[20] G. Belanger, A. Mjallal, and A. Pukhov, “Recasting direct detection limits within micrOMEGAs and implication for non-standard Dark Matter scenarios,” *Eur. Phys. J.* **C81** (2021) no.3, 239, arXiv:2003.08621 [hep-ph].

[21] **XENON** Collaboration, E. Aprile *et al.*, "Constraining the spin-dependent WIMP-nucleon cross sections with XENON1T," *Phys. Rev. Lett.* **122** no. 14, (2019) 141301, arXiv:1902.03234 [astro-ph.CO].

[22] **XENON** Collaboration, E. Aprile *et al.*, "Dark Matter Search Results from a One Tonne×Year Exposure of XENON1T," *Phys. Rev. Lett.* **121** no. 11, (2018) 111302, arXiv:1805.12562 [astro-ph.CO].

[23] **DarkSide** Collaboration, P. Agnes *et al.*, "Low-Mass Dark Matter Search with the DarkSide-50 Experiment," *Phys. Rev. Lett.* **121** no. 8, (2018) 081307, arXiv:1802.06994 [astro-ph.HE].

[24] **PICO** Collaboration, C. Amole *et al.*, "Dark Matter Search Results from the Complete Exposure of the PICO-60 C_3F_8 Bubble Chamber," *Phys. Rev.* **D100** no. 2, (2019) 022001, arXiv:1902.04031 [astro-ph.CO].

[25] **CRESST** Collaboration, A. H. Abdelhameed *et al.*, "First results from the CRESST-III low-mass dark matter program," *Phys. Rev.* **D100** no. 10, (2019) 102002, arXiv:1904.00498 [astro-ph.CO].

[26] T. Sjostrand, S. Mrenna, and P. Z. Skands, "PYTHIA 6.4 Physics and Manual," *JHEP* **0605** (2006) 026, arXiv:hep-ph/0603175 [hep-ph].

[27] G. Belanger, F. Boudjema, P. Brun, A. Pukhov, S. Rosier-Lees, P. Salati, and A. Semenov, "Indirect search for dark matter with micrOMEGAs2.4," *Comput. Phys. Commun.* **182** (2011) 842–856, arXiv:1004.1092 [hep-ph].

[28] H. Zhao, "Analytical models for galactic nuclei," *Mon. Not. Roy. Astron. Soc.* **278** (1996) 488–496, arXiv:astro-ph/9509122 [astro-ph].

[29] D. Maurin, F. Donato, R. Taillet, and P. Salati, "Cosmic rays below z=30 in a diffusion model: new constraints on propagation parameters," *Astrophys. J.* **555** (2001) 585–596, arXiv:astro-ph/0101231.

[30] A. Semenov, "LanHEP — A package for automatic generation of Feynman rules from the Lagrangian. Version 3.2," *Comput. Phys. Commun.* **201** (2016) 167–170, arXiv:1412.5016 [physics.comp-ph].

[31] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr, and B. Fuks, "FeynRules 2.0 - A complete toolbox for tree-level phenomenology," *Comput. Phys. Commun.* **185** (2014) 2250–2300, arXiv:1310.1921 [hep-ph].

[32] F. Staub, "Sarah 4: A tool for (not only susy) model builders," *Comput. Phys. Commun.* **185** no. 6, (2014) 1773–1790, arXiv:1309.7223 [hep-ph].

[33] T. Hahn and M. Perez-Victoria, "Automatized one loop calculations in four-dimensions and D-dimensions," *Comput. Phys. Commun.* **118** (1999) 153, [arXiv:hep-ph/9807565 [hep-ph]].

[34] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, “micrOMEGAs_3: A program for calculating dark matter observables,” *Comput. Phys. Commun.* **185** (2014) 960–985, arXiv:1305.0237 [hep-ph].

[35] A. Goudelis, “micrOMEGAs: Hands-on session,” https://lapth.cnrs.fr/micromegas.