High-Energy Physics/Cosmology Tools Bootcamp
Dartmouth- TRIUMF- University of Washington
October 27, 2017

MicrOMEGAs
A code for calculation Dark Matter signals in generic mode of particle interaction

Alexander Pukhov
Skobeltsyn Institute of Nuclear Physics.
Moscow, Russia.

Plan of presentation

- General characteristics
- Installation of micrOMEGAs

  Compilation of micrOMEGAs routines

  Compilation of model routines and generation of executable file

  Run-time compilation of libraries of matrix elements
- File structure of micrOMEGAs
- Structure of model directory
- Structure of main.c file
- Discrete symmetry and Dark Matter
- Example: IDM

  Inert Doublet Model

  model files

  micromegas session

Plan of presentation. Continue

- Implementation of new models
- MicrOMEGAs routines for Interface with model of particle interaction
- MicrOMEGAs routines for calculation DM signals

    Relic density

        One component DM

        Two components DM

    Direct Detection

        Effective operator

        Operator expansion

        Form factors of light quarks

        Heavy quark loops

        Nucleon amplitudes

        Nuclei recoil energy

Plan of presentation. Continue

- Indirect detection

  Spectra of stable particles produced in DM annihilation

  Fluxes at Earth.

  Loop induced DM,DM → γ,γ and γ,Z
- Neutrino telescope
- SLHA interface for MSSM-like models
- Interface with Lilith/HiggsBounds/SMODELS
- Particle widths and cross sections
- Plots
- Parallel calculations with micrOMEGAs

General characteristics

Operation system Linux or Darwin.

Language C (C99).
Own code size 14Mb
Included packages :
CalcHEP for matrix element generation
LanHEP for model generation
LoopTools for Dm,Dm → gamma, gamma(Z)
SuSpect, NMSSMTools, CpsuperH spectrum calculation for MSSM-like models
Lilith - for Higgs physics
SMODELS - for collider analyses
All together 185Mb
Downloaded in runtime:
HiggsBounds/HiggsSignals for Higgs physics

Needed compilers: gcc , gfortran
Language for user main code: C/C++/Fortran

Installation of micrOMEGAs package

micrOMEGAS site

http://lapth.in2p3.fr/micromegas

Click *Download and Install* *(left -top part of the screen)*
And then *DOWNLOAD* *(right-top part of the screen)*
The name of received file should be

micromegas_4.3.5.tgz

Unpack it by **tar -xvzf micromegas_4.3.5.tgz**

It should create directory **micromegas_4.3.5/** which occupies about 180 Mb of disk space. You will need more disk space after compilation of specific models and generation of matrix elements .

Compilation of micromegas code consists of 3 steps.

1) Compilation of CalcHEP and main micrOMEGAs routines
2) Compilation of code for given model of particle interaction.
3) Runtime compilation of external packages and matrix elements.

Compilation of micrOMEGAs routines

To compile CalcHEP and main micrOMEGAs routines call

make

In micrOMEGAS directory.

If you would like to use CalcHEP GUI sessions and plots generated by micrOMEGAs be sure that X11 develop package in installed. Namely you have to check existence of X11 header files. They should be disposed in

/usr/include/X11

To install them

libX11-devel for Fedora/Scientific, Darwin(MAC)
libX11-dev for Ubuntu/Debian
xorg-x11-devel for SUSE

make clean

deletes all generated files in main directories and model directories

Compilation of specific model routines and generation of executable *main* file.

All model directories contain

main.c and Makefile

Command

make

generates executable

./main

The user can modify main.c or write his own *my_main.c*

User main program can be compile by the command

make main=*my_main.c*

Which generates executable file

./my_main

Default main.c codes disposed in micrOMEGAs model directories generate executable which needs one parameter, a name of file with numerical values of model parameters.

To execute

./main data.par

Run time compilation of matrix elements

If micrOMEGAs needs matrix element for some process, or structure of model vertex it calls CalcHEP for matrix element generation. Code of matrix element/vertex is compiled, presented as shared library and stored in directories

MODEL/work/so_generated.

User sees the message on the screen

PROCESS: <name of process>

Or

VERTEX : < name of vertex>

Information about vertices is used to compile effective loop induced Higgs -photon and Higgs-gluon vertices.

Shared library is loaded dynamically in run time

Shared libraries generated only one time.

If model of interaction is changed, then shared library recompiled automatically

File structure of micrOMEGAs package.

micromegas_4.3.5/ main directory

CalcHEP_src/ generator of matrix elements
sources/ micrOMEGAs own codes
man/ description of micrOMEGAs routines
manual_4.2.tex, manual_4.2.pdf
Include/
micromegas.h & micromegas_aux.h
lib/
Packages/
SuSpect_2.41 NMSSMTools_4.7.1 CpsuperH2.3, LoopTools-2.1 LanHEP

model directories:

MSSM/
NMSSM/ Next-to-Minimal SuSy Model
CPVMSSM/ MSSM with complex parameters
UMSSM/ MSSM + U(1) gauge field
IDM/ Inert dublet model
LHM/ Little Higgs Model
Z3IDM/ Z^3 model
Z4IDMS/ Z^4 model

Structure of MODEL directory

Makefile

main.c main.F files with *main* program for given model

lib/

***.c, .F,cpp** source codes of specific model routines

Makefile called automatically to generate

alib.a compiled library

work/ CalcHEP working directory intended for matrix element generation

models/ model in CALCHEP format

vars1.mdl **func1.mdl** **prtcls1.mdl** **lgrng1.mdl** **extlib1.mdl**

so_generated/ directory to store automatically generated matrix elements

calchep/ for interactive CalcHEP sessions

Makefile supports compilation of C,Fortran and C++ user codes

[g]make main=XXX.c => executable XXX

[g]make main=YYY.F => executable YYY

[g]make main=ZZZ.cpp => executable ZZZ

[g]make is equivalent to [g]make main=main.c

Structure of main.c file

main.c, main.F main.cpp files presented in micrOMEGAs model directories consist from several independent blocks enclosed into

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

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

```
#define MASSES_INFO                 // Display information about mass spectrum
#define CONSTRAINTS                 //   Display B->s,gamma, Bs->mu,mu,
#define LILITH                      //  Test of  Higgs properties
#define HIGGSBOUNDS
#define OMEGA                       // Calculate relic density
#define INDIRECT_DETECTION          // Signals of DM annihilation in galaxy hallo
//#define RESET_FORMFACTORS         // Redefinition of  Form Factors and other parameters
#define CDM_NUCLEON                 // Calculate amplitudes and cross-sections for  CDM-
                                        nucleon  collisions
//#define CDM_NUCLEUS               // Calculate number of events for 1kg*day and recoil
                                    //  energy distribution  for various nuclei
#define NEUTRINO                    // neutrino telescope
#define  DECAY                      // particle width and decay branching
#define  CROSS_SECTIONS             // calculate cross sections
```

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

Dark Matter in micrOMEGAs models.

Discrete symmetry.

MicrOMEGAs assumes a discrete symmetry which is responsible for stability of Dark Matter. For instance, it could be a Z_2 symmetry which divides all particles in two classes, odd and even, say R-parity in MSSM. The lightest odd particle is stable and can be treated as DM.

For micrOMEGAs odd particles are particles whose name started with tilde "~". For example, ~X,~H3,~H+ in IDM.

In case of Z_4 symmetry internal charge for DM particles can be +/- 1 or 2. DM1- the lightest particle with charge 1 is always stable. But the lightest particle with charge 2 in stable if its mass less then mass of 2 DM1 particles. One can also construct a model with complex symmetry like $Z_2 \times Z_3$ which always has 2 DM particles.

MicrOMEGAs can work with models with 2DM classes which are marked by "~" and "~~"

Example: Inert Doublet Model

Inert Doublet model contains two SU(2)*U(1) doublets

$$H_1 = \begin{pmatrix} 0 \\ <v> + h/\sqrt{2} \end{pmatrix} \quad , \quad H_2 = \begin{pmatrix} \tilde{H}+ \\ (\tilde{X} + i \cdot \tilde{H}3)/\sqrt{2} \end{pmatrix}$$

The Lagrangian contains only *even* powers of H_2 doublet

$$L = (SM\ terms) + D^{\mu} H_2^* D_{\mu} H_2$$

$$-\mu^2 H_2^2 - \lambda_2 H_2^4 - \lambda_3 H_1^2 H_2^2 - \lambda_4 |H_1^* H_2|^2 - \lambda_5 Re[(H_1^* H_2)^2]$$

Because of symmetry $H_2 \rightarrow -H_2$ the lightest of $\tilde{H}^+, \tilde{X}, \tilde{H}3$ is stable

Parameters $\mu,\ \lambda_3,\ \lambda_4$ can be expressed in terms of masses

New couplings are λ_2 , $\lambda_L = \lambda_3 + \lambda_4 + \lambda_5$

See details **arXiv:1106.1719**

vars1.mdl: Free parameters of the model.

```
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
.............................
```

func1.mdl: Constrained parameter of the model.

```
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
```

prtcls1.mdl: Particles of the model

List fo particles presented in file MODEL/work/models/prtcls1.mdl

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)

Names of particles of odd sector start with tilde ~

lgrng1.mdl: Feynman rules

Inert Dublet
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

…..............

p – momentum, m – Lorentz index

Example of micrOMEGAs session for IDM

./main data1.par

```
VERTEX: W- W+ h
VERTEX: L l h
VERTEX: C c h
VERTEX: T t h
VERTEX: B b h
VERTEX: ~H- ~H+ h

Dark matter candidate is '~X' with spin=0/2

=== MASSES OF HIGGS AND ODD PARTICLES: ===
Higgs masses and widths
PROCESS: h->2*x
PROCESS: W+->2*x
PROCESS: Z->2*x
PROCESS: h->W-,E,ne
Delete diagrams with W+<1
PROCESS: h->Z,ne,Ne
Delete diagrams with Z<1
     h   125.00 3.97E-03

Masses of odd sector Particles:
~X   : MHX   =   600.0 || ~H3  : MH3   =   601.0 || ~H+  : MHC   =   604.0

LILITH(DB15.09):  -2*log(L): 25.96; -2*log(L_reference): 0.00; ndf: 38; p-value: 9.31E-01
```

Continue

==== Calculation of relic density =====
PROCESS: ~X,~X ->AllEven,1*x{A,Z,G,W+,W-,ne,Ne,e,E,nm,Nm,m,M,nl,Nl,l,L,u,U,.....
PROCESS: ~H3,~X ->AllEven,1*x{A,Z,G,W+,W-,ne,Ne,e,E,nm,Nm,m,M,nl,Nl,l,L,u,U,...
PROCESS: ~H3,~H3->AllEven,1*x{A,Z,G,W+,W-,ne,Ne,e,E,nm,Nm,m,M,nl,Nl,l,L,...
…….....

Xf=2.62e+01 Omega=1.13e-01
# Channels which contribute to 1/(omega) more than 1%.
# Relative contributions in % are displayed
21% ~X ~X ->W+ W-
14% ~X ~X ->Z Z
11% ~H3 ~H3 ->W+ W-
9% ~H+ ~H- ->W+ W-
7% ~H3 ~H3 ->Z Z
6% ~H+ ~X ->A W+
5% ~H3 ~H+ ->A W+
4% ~H+ ~H- ->A A
4% ~H3 ~H+ ->Z W+
3% ~H+ ~X ->Z W+
3% ~H+ ~H- ->A Z
2% ~H+ ~H- ->Z Z
2% ~H+ ~X ->W+ h
1% ~H+ ~H- ->h h

```
==== Indirect detection =======
 annihilation cross section 6.18E-26 cm^3/s
 contribution of processes
  ~X,~X -> W+ W-       6.01E-01
  ~X,~X -> Z Z         3.99E-01
sigmav=6.18E-26[cm^3/s]
Photon flux  for angle of sight f=0.10[rad]
and spherical region described by cone with angle 0.10[rad]
Photon flux = 9.37E-16[cm^2 s GeV]^{-1} for E=300.0[GeV]
Positron flux  =  1.04E-13[cm^2 sr s GeV]^{-1} for E=300.0[GeV]
Antiproton flux  =  5.91E-13[cm^2 sr s GeV]^{-1} for E=300.0[GeV]


==== Calculation of CDM-nucleons amplitudes  =====
PROCESS: QUARKS,~X->QUARKS,~X{u,U,d,D,c,C,s,S,t,T,b,B
Delete diagrams with _S0_!=1,_V5_,A
CDM[antiCDM]-nucleon micrOMEGAs amplitudes:
proton:  SI  1.497E-11 [1.497E-11]  SD  0.000E+00 [0.000E+00]
neutron: SI  1.512E-11 [1.512E-11]  SD  0.000E+00 [0.000E+00]
CDM[antiCDM]-nucleon cross sections[pb]:
 proton  SI 9.767E-14 [9.767E-14] SD 0.000E+00 [0.000E+00]
 neutron SI 9.962E-14 [9.962E-14] SD 0.000E+00 [0.000E+00]
==============Neutrino Telescope=======  for  Sun
 E>1.0E+00 GeV neutrino/anti-neutrino fluxes   1.81E+01/2.05E+01 [1/Year/km^2]
IceCube22 exclusion confidence level = 1.29E-07%
 E>1.0E+00 GeV Upward muon flux    2.337E-07 [1/Year/km^2]
 E>1.0E+00 GeV Contained muon flux 6.999E-07 [1/Year/km^3]
```

Implementation of new models in micrOMEGAs

• The command

./newProject *MODEL*

launched from the root micrOMEGAs directory creates the directory *MODEL*, which contains all files needed to run micrOMEGAs (for example main.c) with the exception of the new model files.

• The new model files in the CalcHEP format should then be included in the sub-directory MODEL/work/models. The files needed are

vars1.mdl, func1.mdl, prtcls1.mdl, lgrng1.mdl extlib1.mdl

Simple example:

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

It should work!

Implementation of new models:
Generation of model files in CalcHEP Format

Model files can be created by mean of
LanHEP, FeynRules, Sarah

LanHEP is included in micrOMEGAs package. Each model directory contains lanhep subdirectory with source files with Makefile which calls LanHEP.

See LanHEP manual
micromegas_X.Y/Packages/LanHEP/manual/man31.pdf

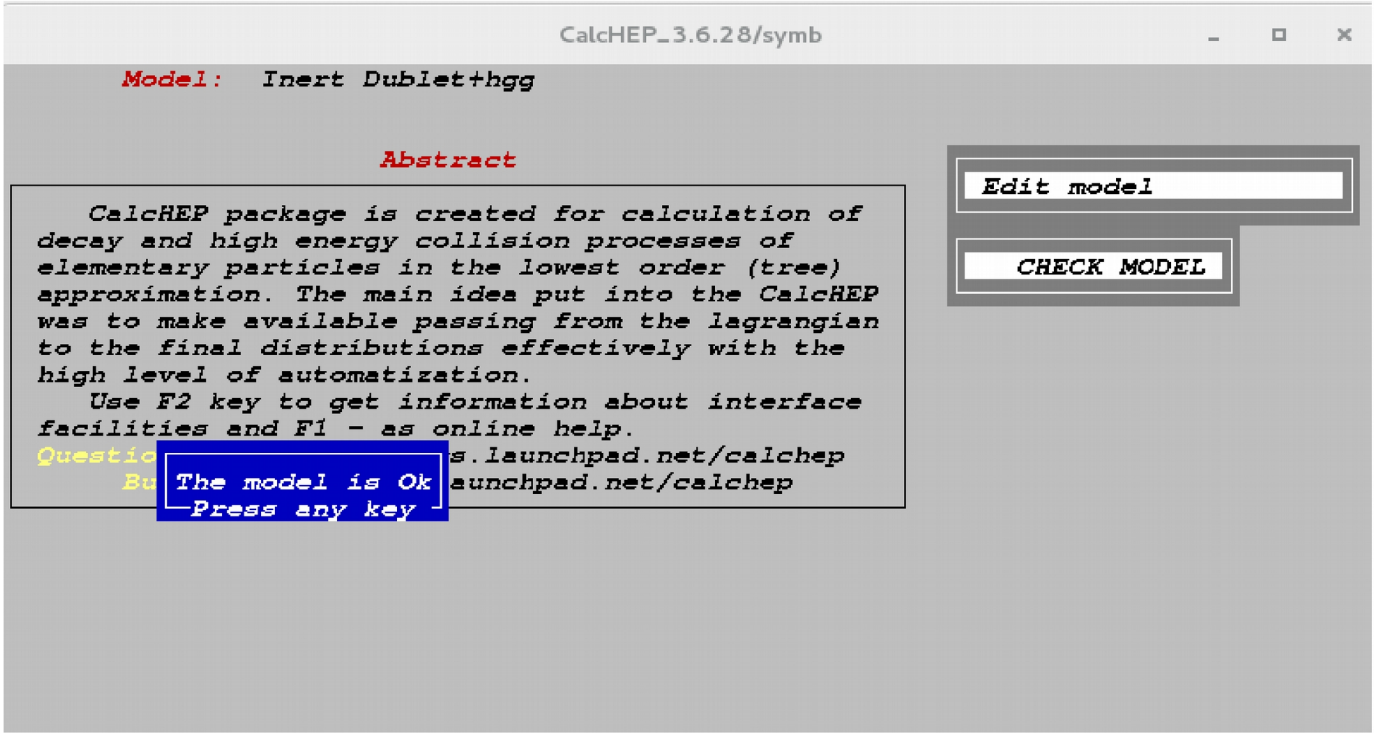
Follow examples presented in any micrOMEGAs model
The simplest one is in IDM/lanhep

Testing of created model

make command launched from the model directory checks the model and stops with error code if model files do not correspond to CalcHEP requirements.

The user can go to work directory, launch ./calchep and use menu line

CHECK MODEL



Free model parameters.

.

assignValW(*name,value*) assigns new value to parameter.

For example

assignValW(“MHX”,600)

Function to download set of parameters:

readVar(*fileName***)**

Structure of file has to be

name value [# comment]

For instance, in case of IDM

```
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
```

Checking of constrained models parameters

After assignment of parameters one has to call

sortOddParticles(outText);

which calculates constrained parameters and finds DM particle[s]. In case of error in calculation of constrained parameter this routine returns error code and *outText* contains the name of parameter which can not be calculated.

In case of success sortOddParticles detects the lightest odd particle[s]

CDM1 [CDM2] and their masses

Mcdm1 [Mcdm2, Mcdm=min(Mcdm1,Mcdm2)]

Values of constrained parameters can be obtained by

findValW(*name*)

Masses of particles can be obtained by

pMass(*name*)

Calculation of DM relic density for 1 DM case

darkOmega(&Xf,fast,Beps) uses Runge-Kutta to solve

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

darkOmegaFO(&Xf,fast,Beps) uses freeze-out approximation

Both return Ωh^2

fast =*1* *for fast* calculation: Gauss n-point integration with separation of s-channel poles

0 for precise calculation , Simpson adaptive integration

-1 for very robust Simpson adaptive integration with separation of s-channel poles

Beps removes co-annihilation if

$$exp(\frac{2M_{cdm} - M_1 - M_2}{T}) < Beps$$

X_f =Mcdm/T_f , where $Y(T_f) = 2.5\, Y_{eq}(T_f)$ defines freeze-out temperature

printChannels(Xf, Beps, cut,prc,file) prints out main annihilation channels and their contributions to 1/ Ωh^2

21% ~X ~X ->W+ W-
14% ~X ~X ->Z Z
11% ~H3 ~H3 ->W+ W-
9% ~H+ ~H- ->W+ W-

Relic density for 2-components Dark Matter

darkOmega2(fast,Beps) returns Ωh^2

If Mdm1 and Mcdm2 are different we first have freeze out for heavy DM. The lightest one is in thermal equilibrium with SM particles and returns fast to equilibrium state in case of any deviation.

$$\frac{d\Delta Y}{ds} = \frac{2Y_{eq} < v\sigma >}{3H}\Delta Y + ...$$

Thus Runge-Kutta needs very small step for solution

Special stiff solution (Numerical Recipes in C) is applied.

No printChannels for darkOmega2

Direct Detection

To predict results of direct detection experiment in a given model we have to calculate cross sections of DM – nuclei elastic scattering.
The model defines

DM - quarks interaction

Then we have to calculate

DM - nucleon scattering cross section

And at next step

DM -nuclei scattering cross section

Velocities of DM particles in halo of Milky Way are about orbital velocities of stars

$$v \approx 220 km/s \approx 10^{-3} c$$

One can treat such scattering as a scattering at v → 0 limit, taking into account that elastic cross section has a finite value at this limit.

DM – quark/nucleon interaction in the v → 0 limit

Can be described in terms of effective operators. There are 4 types of such operators

SI – Spin independent (scalar) – interactions without spin flip.
SD – Spin dependent – interactions with spin flip.
Even - DM and DM* have the same amplitude.
Odd - DM and DM* amplitudes have different signs.

Operator expansion

MicrOMEGAs extends given model of particle interaction adding such effective operators

$$\hat{\mathcal{L}}_{eff}(x) = \sum_{q,s=(even,odd)} \lambda_{q,s}\hat{\mathcal{O}}_{q,s}(x) + \xi_{q,s}\hat{\mathcal{O}}'_{q,s}(x)$$

and finds cofficients at this operators calculating amplitudes for collision at rest

$$\langle q(p_1), \chi(p_2)|\hat{S}\hat{\mathcal{O}}_{e,o}|q(p_1), \chi(p_2)\rangle$$

$$\langle \bar{q}(p_1), \chi(p_2)|\hat{S}\hat{\mathcal{O}}_{e,o}|\bar{q}(p_1), \chi(p_2)\rangle$$

Nucleon form factors for light quarks

Each operator at quark level leads to the same type operator a nucleon level with form factor

Even scalar form factors

The operator $\langle N|m_q\overline{\psi}_q\psi_q|N\rangle$ is interpreted as the contribution of quark q to the nucleon mass, M_N,

$$\langle N|m_q\overline{\psi}_q\psi_q|N\rangle = f_q^N M_N$$

$f_{u,d,s}^N$ are known from hardon spectroscopy, data of πN scattering and lattice calculations (s-quark)

Odd scalar form factors

$$\langle N|\overline{\psi}_q\gamma_\mu\psi_q|N\rangle = \mathrm{f}_{V_q}^N\langle N|\overline{\psi}_N\gamma_\mu\psi_N|N\rangle$$

Just give us (quark) - (anti-quark) number counting because of vector current conservation.

$$f_{V_u}^P = 2 \quad f_{V_d}^P = 1$$

Even vector form factor $\gamma_5\gamma_\mu$

Describe contribution of quarks and and anti-quarks to nucleon spin

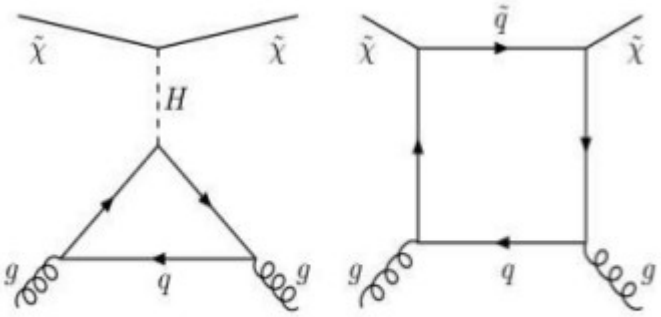
Odd vector form factor $\sigma_{\mu\nu}$

Describe difference of contribution of quarks and antiquarks to nucleon spin.

Form factors of light quarks are presented by global parameters

Proton		Neutron		
Name	value	Name	value	comments
ScalarFFPd	0.0191	ScalarFFNd	0.0273	
ScalarFFPu	0.0153	ScalarFFNu	0.011	Scalar form factor
ScalarFFPs	0.0447	ScalarFFNs	0.0447	
pVectorFFPd	-0.427	pVectorFFNd	0.842	
pVectorFFPu	0.842	pVectorFFNu	-0.427	Axial-vector form factor
pVectorFFPs	-0.085	pVectorFFNs	-0.085	
SigmaFFPd	-0.23	SigmaFFNd	0.84	
SigmaFFPu	0.84	SigmaFFNu	-0.23	Tensor form factor
SigmaFFPs	-0.046	SigmaFFNs	-0.046	

Heavy quark loops



Diagrams that contribute to DM-gluon interaction via heavy quark loops

Heavy quarks interact with nucleon via gluon condensate. For triangle (Higgs) heavy quark condensate is a good approximation. For box diagrams one needs loop calculation.

For renormalizable interactions corresponding boxes where presented in

DM spin 1/2 M.Drees & M.Nojiri hep-ph/9307208
DM spin 0 and 1 Hisano,Junji,Nagai,Ryo,Nagata,Natsumi arXiv:1502.02244

MicrOmegas replaces propagators on corresponding loop functions without testing type of interaction arXiv 0803.2360

Nucleon amplitudes and cross sections in micrOMEGAs

nucleonAmplitudes(name_of_DM ,pA0,pA5,nA0,nA5);
Output: *pA0,pA5,nA0,nA5* – 2 dimension arrays

Proton

pA0[even SI, odd SI] pA5[even SD, odd SD]

Neutron

nA0[even SI, odd SI] nA5[even SD, odd SD]

Then DM-nucleon cross section in [pb] units are

$\sigma_{si} = C*A^2$ $\sigma_{SD} = 3*C*A^2$ where $C=4/\pi*3.89E8*(M_N*M_{dm}/(M_N+M_{dm})^2$

This cross section is calculated by micrOMEGAs

CDM[antiCDM]-nucleon micrOMEGAs amplitudes:
proton: SI 1.497E-11 [1.497E-11] SD 0.000E+00 [0.000E+00]
neutron: SI 1.512E-11 [1.512E-11] SD 0.000E+00 [0.000E+00]

and can be directly compared with values extracted from experiments

Nuclei interactions

Nuclei form factors

For zero DM velocity DM-nucleus SI cross section reads

$$\sigma_0^{SI} = \frac{4\mu^2}{\pi}\left(\lambda_p Z + \lambda_n (A-Z)\right)^2 \ , \quad \mu = \frac{M_{cdm} M_A}{M_{cdm} + M_A}$$

where λ_p, λ_n are amplitudes for DM scattering on nucleons; M_A, Z, A are the nucleus mass, charge, and atomic number respectively. For a small DM velocity, $v \approx 10^{-3}c$, we neglect the dependence on the small momentum transfer in the cross section but include this dependence in the nucleus form factor

$$\frac{d\sigma^{SI}}{dE} = \frac{\sigma_0^{SI}}{E_{max}} F_A^2(q) \ , \quad 0 < E < E_{max} = 2\left(\frac{v^2\mu^2}{M_A}\right)$$

For SI interactions, $F(q)$ is a Fourier transform of the nucleus distribution function,

$$F_A(q) = \int e^{-iqx} \rho_A(x) d^3x$$

micrOMEGAs use the Fermi distribution function

$$\rho_A(r) = \frac{c_{norm}}{1 + exp((r - R_A)/a)}$$

where $a = 0.52fm$ nuclei surface thickness, and
$R_A = 1.23A^{\frac{1}{3}} - 0.6$fm nuclei radius

There are similar but more complicated formulas for SD nucleus cross section which depends on 3 form factors, proportion to nucleus momentum J.

SD interaction does not lead to A enhancement.

micrOMEGAs function for nuclei

nucleusRecoil(
  f, - velocity distribution f(v[km/s]) normalized by $\int_0^\infty v f(v) dv = 1$
  A, - atomic number
  Z, - nucleus charge
  J, - number of spin states
  Sxx, - SD formfactors
  dNdE - recoil energy distribution stored in array
)

dNdERecoil(E[keV],dNdE) interpolates dNdE table and gives spectrum in 1/keV/kg/day units

For example:

```
nEvents=nucleusRecoil(Maxwell,73,Z_Ge,J_Ge73,SxxGe73,dNdE);
          Result depends on global parameter
 rhoDM    0.3[GeV/cm^3] Dark Matter density at Rsun
 Vrot     220[km/s]     Galaxy rotation velocity at Rsun
 Vearth   225[km/s]     Galaxy velocity of the Earth
 Vesc                   Escape velocity at Rsun
```

Indirect detection in micrOMEGAs

Indirect detection -detection of photons, positrons and antiprotons signal obtained in result of DM annihilation in Galactic Hallo.

For various spectra we use NZ=250 dimention arrays and interpolation function for them is SpectdNdE(E,spectArr)
One can use displayPlot to see and compare different spectra.

vsigma=calcSpectrum(key,Sg,Se,Sp,Sne,Snm,Snl,&err)
Calculates vσ cross section in cm^3/sec units for DM annihilation photon Sg, positron Se, antiprotons Sp, and neutrino Sne,Snm,Snl contain spectra for DM-DM annihilation
Here the avarage over DM,DM/antiDM is done. dmAssym is taken into account, In case of 2 DM particles we have an average over all types of collisions. PYTHIA 6.4 was used for hadronisation of primary annihilation channels.

Meaning of key parameter: 1-takes into account W/Z polarization
2-include gammas from 2->2+gamma
4-print cross sections

calcSpectrum

main.c record:

```
sigmaV=calcSpectrum(4,SpA,SpE,SpP,SpNe,SpNm,SpNl ,&err);
```

MicrOMEGAs output

```
annihilation cross section 6.18E-26 cm^3/s
contribution of processes
  ~X,~X -> W+ W-       6.01E-01
  ~X,~X -> Z Z         3.99E-01
```

The cross section for given channels can be compared with limits obtained by FermiLAT

Spectra for gamma, positrons antiprotons and neutrinos are obtained by PYTHIA and stored in
SpA, SpE, SpP, and SpN[e,m,l] arrays.
Numerical values for dN/dE for given energy E can be obtained by

```
spectdNdE( E, SpX)
```

To get particle fluxes at Earth level one has for take into account DM distribution in Galaxy and propagation charged particles in Galactic magnetic field.

loopGamma

For all implemented models we have
DM,DM-> photon, photon and DM,DM -> photon, Z
loop induced signals. These signals are not compiled automatically in run-time but generated in advance by means of FormCalc.

One has to uncomment
//#define LoopGAMMA
to force micrOMEGAs to calculate point like gamma signal.
Function loopGamma(&vcs_gz,&vcs_gg) calculates annihilation rates vcs_gz and vcs_gg [cm^3/s]. For example for IDM model with data1.par parameters

```
Gamma ray lines:
E=5.97E+02[GeV]  vcs(Z,A)= 1.58E-28[cm^3/s], flux=4.91E-14[cm^2 s]^{-1}
E=6.00E+02[GeV]  vcs(A,A)= 5.37E-29[cm^3/s], flux=3.33E-14[cm^2 s]^{-1}
```

Calculated cross sections can be compared with experiments for search of single lines in photon spectrum

The loopGamma function is not available automatically for models implemented by the user.

Halo profile

DM distribuion is defined by DM density at Sun, parameter rhoDM and halo profile. By default micrOMEGAs uses Zhao profile

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

with alpha=1, beta=3 rc=20kpc.

setProfileZhao(α,β,γ,rc) change these parameters.

setHaloProfile(F) allows to substitute any profile presented by function F(r)

The command **setHaloProfile(hProfileZhao)** sets back the Zhao profile

Photon flux

gammaFluxTab(fi,dfi,sigmav,Sg,Sobs)
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 $2\pi(1-\cos(df\ i))$,
sigmav is the annihilation cross section,
Sg - photon spectrum at point of annihilation
Sobs is tabulated photon flux

SpectdNdE(E,Sobs)

gives resulting photon flux in [1/(GeV cm^2 s)] units

gammaFlux(fi,dfi,vcs_gz))
gammaFlux(fi,dfi,2*vcs_gg)
 return corresponding fluxes for loop induced processes

Antiproton and positron fluxes

- **posiFluxTab**(Emin,sigmav, Se, Sobs)

- **pbarFluxTab**(Emin,sigmav, Sp, Sobs)

The same style as for photons. But depends on propagation parameters

```
K_dif     | 0.0112| kpc^2/Myr |The normalized diffusion coefficient
L_dif     |  4    | kpc       | Vertical size of the Halo diffu
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
```

And finally

solarModulation(Phi, mass, stellarTab, earthTab)

allows to take into account solar modulation effect.

Here Phi potential [MeV], mass is mass of particle,
stellarTab flux before modulation
earthTab flux after modulation.

Neutrino telescope

micrOMEGAs uses direct detection module to calculate number of DM captured by Sun/Earth.

Captured DM is concentrated in the center of Sun/Earth and neutrino produced in result of DM annihilation can be detected by neutrino telescope experiment (IceCube, Super-Kamiokande, Baksan).

DM annihilation inside of Sun/Earth is different from annihilation in vacuum. Also there are effects of propagation and oscilation.

For flux of resulting muon neutrinos micrOMEGAs uses tables obtained by WimpSim package: J. Edso et.al arXiv 0709.3898
Or
PPPC4DMnu: M. Cirelli, et.al. arXiv 1312.6408

Agreement between two sets is not very good.

MicrOMEGAs routine basicNuSpectra reads these tables depending on
WIMPSIM flag
WIMPSIM=1 for WimpSIm
WIMPSIN=0 for PPPC4DMnu

basicNuSpectra(*forSun,Mcdm,pdg, pol, nu, nu_bar*)
where

forSun is 1 or 0,

pdg - is PDG number of annihilation channel.

pol=-1(1) corresponds to longitudinal (trans-verse) polarisation of vector bosons or to left-handed (right-handed) fermions, pol=0 is used for unpolarized spectra.

Arrays **nu, nu_bar** contains spectra.

SpectdNdE(E,spect) interpolates arrays.

Combining DM capture rate and annihilation spectra micrOMEGAs calculates muon neutrino fluxes at Earth surface

neutrinoFlux(Maxwell,forSun, nu,nu_bar);

After that one can apply iceCude22 limits for neutrino spectra: iceCube22 arXiv 0902.2460

exLevIC22(nu,nu_bar,NULL) exclusion level.

MicrOMEGAs is able to calculate muon spectra produced to neutrinos, but we have not now angular resulution for muon flax. It should be improved to apply micrOMEGAs to other neutrino telescope experiments

Dark Matter Asymmetry

If DM particles is not self-conjugated one can assume Dm- antiDm asymmetry similar to barion asymmetry. In micrOMEGAs gobal parameter

deltaY

presents difference between DM/anti-DM abundances. darkOmega[FO] takes it into account. darkOmega2 - not

dmAsym parameter is calculated

$\Omega(+/-) = \Omega(1 +/- dmAsym)/2$

dmAsym contributes to all function of direct/indirect detection and neutrino telescope.

DarkOmega2 does not take into account deltaY parameter.

SLHAplus[arXiv 1008.0181]: Tools for MSSM-like models

MSSM-like models need external program for calculation of particles spectrum. SLHA file exchange interface was developed for it.

Routines **slhaRead, slhaVal openAppend, aPrintF**

File with particle spectrum | **CalcHEP model file**

```
BLOCK MASS    # Mass spectrum                  |  slhaRead(file_name, mode)
#  PDG Code      mass                 particle |
 25    1.15137179E+02   # neutral Higgs      | Mh   | slhaVal("MASS",0,1,25)
 37    1.48428409E+03   # charged Higgs      | MHC  | slhaVal("MASS",0,1,37)
                                               |
BLOCK NMIX   # Neutralino Mixing Matrix       |
 1   1      9.98499129E-01    # Zn11          | Zn12 | slhaVal("NMIX",0,2,1,2)
 1   2     -1.54392008E-02    # Zn12          | Zn12 | slhaVal(“NMIX”,0,1,1,2)
```

In main.c can choose the external code to compute the spectrum or read spectrum from SLHA file

```
#define RGE  suspect
    /* choose 'suspect','isajet','softSusy','spheno'*/
/*=========  SUSY scenario  ==========
  One can define SUGRA, AMSB, EWSB (for low scale input).
  By default the program reads SLHA data file
============================================*/
//#define SUGRA
//#define SUGRANUH
//#define AMSB
#define EWSB
```

SLHAplus interface for HiggsBounds/Lilith/SMODELS

micrOMEGAS uses files for interface with external packages. Output of HiggsBounds/Lilith/Smodels is presented in extended SLHA format. SLHAplus library was updated correspondingly For example

```
Block HiggsBoundsResults
#CHANNELTYPE 1: channel with the highest statistical sensitivity
 1  1        328                                                    # channel id
 1  2          1                                                     # HBresult
 1  3   0.7269279334500290                                         # obsratio
 1  4          1                                                     # ncombined
 1  5   ||(p p)->h+..., h=1 where h is SM-like (CMS-PAS-HIG-12-008)|| # channel
```

```
slhaSTRFormat("HiggsBoundsResults","1 5 ||%[^]]||", channel);
```

```
Block FOBS # Flavour observables
# ParentPDG type value      q   NDA ID1 ID2 ID3 ... comment
   5     1   2.95061156e-04  0    2   3  22       # BR(b->s gamma)
 521     4   8.35442304e-02  0    2 313  22       # Delta0(B->K* gamma)
 531     1   3.24270419e-09  0    2  13 -13       # BR(B_s->mu+ mu-)
```

```
Bsg= slhaValFormat("FOBS", 0., "5 1 %E 0 2 3 22")
```

Particle width and decay branching can be obtained by

```
pname = "h";
numout*L;
width=pWidth(pname,&L);
printf("\n%s :   total width=%.2E \n and Branchings:\n",pname,width);
printTxtList(L,stdout);
```

```
h :   total width=4.056E-03
 and Branchings:
2.347674E-02 h -> Z,Z
2.020447E-01 h -> W+,W-
2.311262E-03 h -> A,A
7.871916E-02 h -> G,G
2.252752E-04 h -> m,M
6.359358E-02 h -> l,L
6.049041E-06 h -> u,U
6.049041E-06 h -> d,D
2.851687E-02 h -> c,C
2.419579E-03 h -> s,S
5.986808E-01 h -> b,B
```

VZdecay and VWdecay flags switch on/off virtual Z/W decays both in width calculation and DM annihilation

If micromegas reads decay information from an SLHA file then ***pWidth*** *returns the width presented in the file.*

Cross sections

numout *cc ; // numout – is a type for matrix element in micrOMEGAs.

cc = newProcess(char*Process); // call CalcHEP to calculate symbolically and compile matrix element for given process. For instance

cc = newProcess("e,E->m,M");

Cross sections of 2->2 processes can be calculated by

cs= cs22(cc,l,Pcm,cos_min,cos_max,&err);

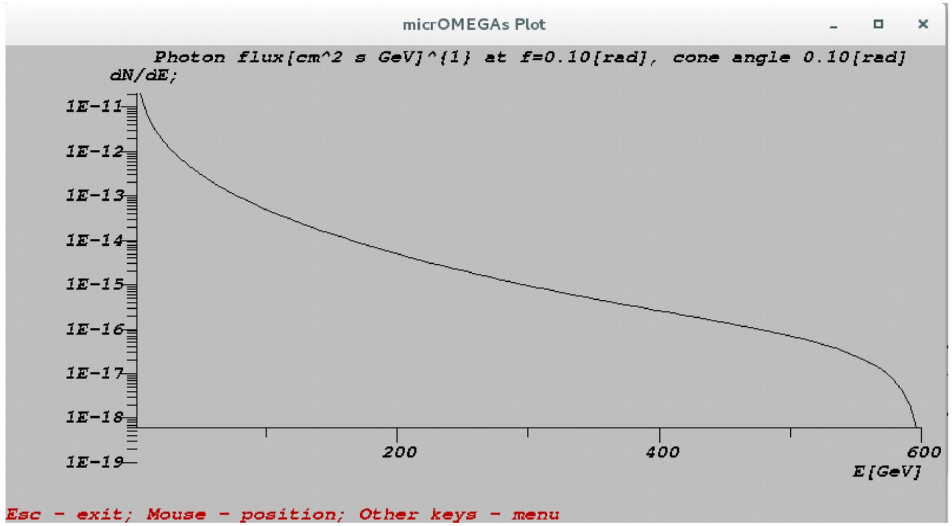
Pcm – momentum in Center of Mass reference frame

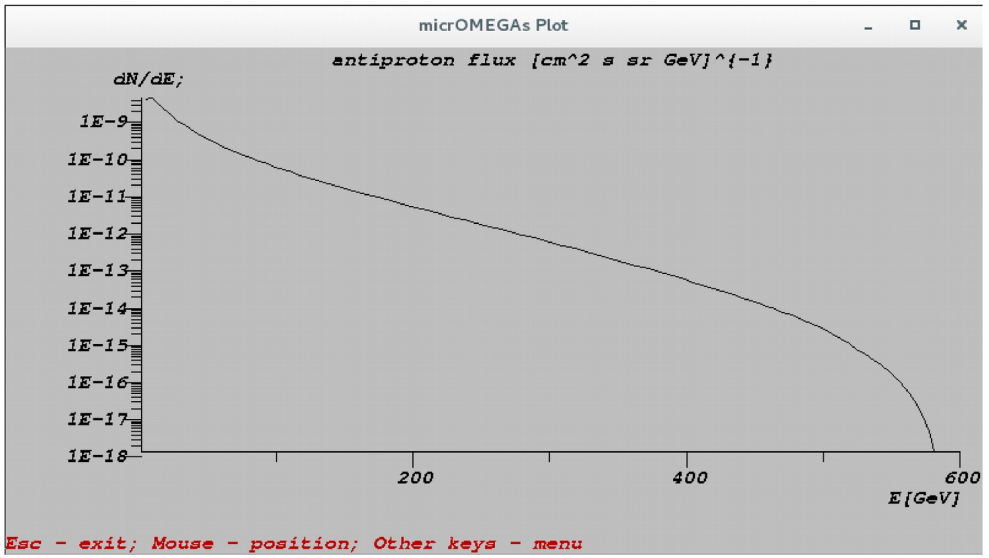
cos_min, cos_max - cuts for cosine of scattering angle in the same frame

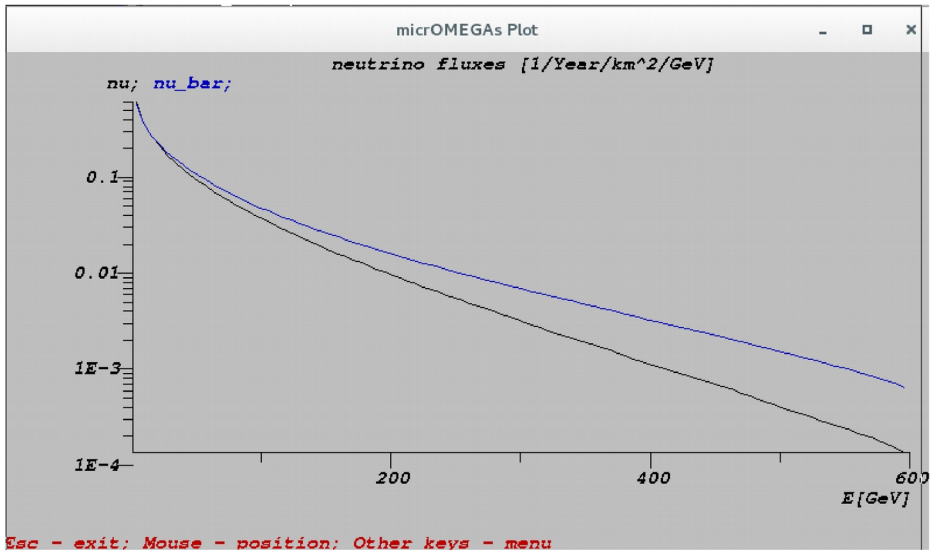
l=1 in case you have generated codes only for one process. For general case l numerates subprocesses.

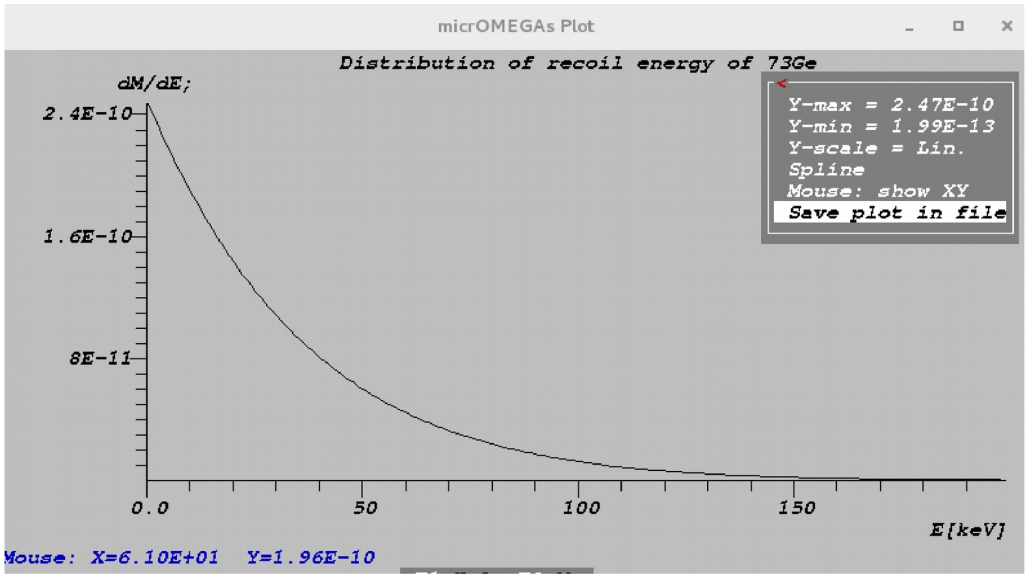
Plots

One can uncomment #define SHOWPLOTS in main.c
To get plots produced by micrOMEGAS









The plots can be saved in Root, PAW, Gnuplot formats

Parallel calculations in micrOMEGAs

I have not experience but have suggestions!

Because of micrOMEGAs uses file interface with external packages one can not launch several sessions in the same place of disk space.

But micromegas executable file uses only absolute paths to data files and external executable files. So, one can copy executable file in any place and it will work correctly. It is better to use symbolic link for such coping

ln -s main ../other_directory

./main launched parallel in different directories will work almost independently. But

1) one has to pass then different input files or randomize random number generator to prevent identical parallel operation.

2) different ./main programs will create shared libraries in the same place

mother_directory/work/so_generated

MicrOMEGAs foresees it.

a) Each process creates its own subdirectory for CalcHEP session

b) Each process creates a 'lock' file before writing down shared library.

The following trick was proposed for safety parallel generation of shared libraries: before parallel calculation call darkOmega with parameter Beps=1.

micrOMEGAs should work correctly without this trick.