IFIC
INSTITUT DE FÍSICA CORPUSCULAR

LanHEP and MicrOmegas

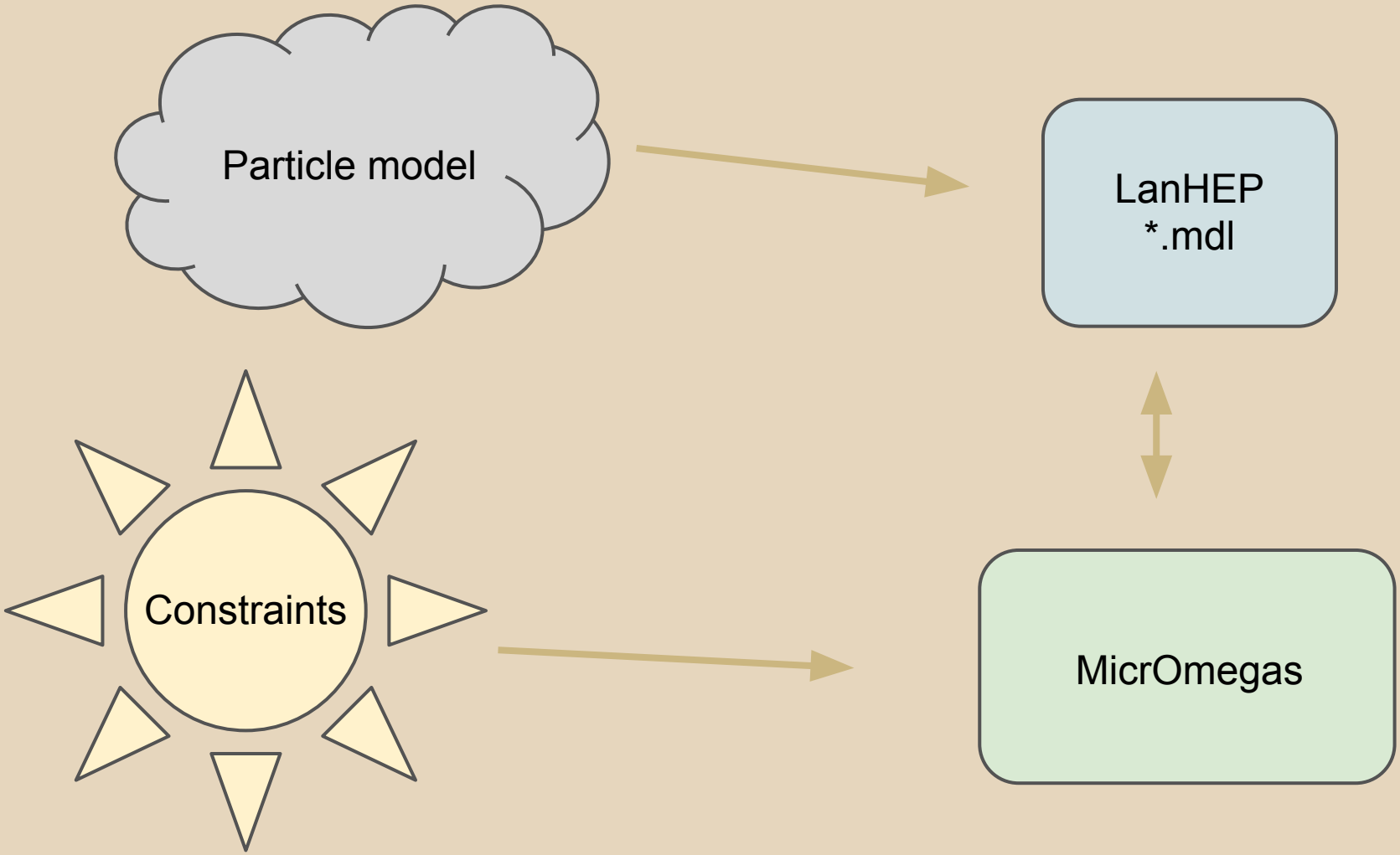
R. Lineros and J. Palacio

Workshop on DM tools and Hands-on Fermi analysis tools
22-26 April. Valencia

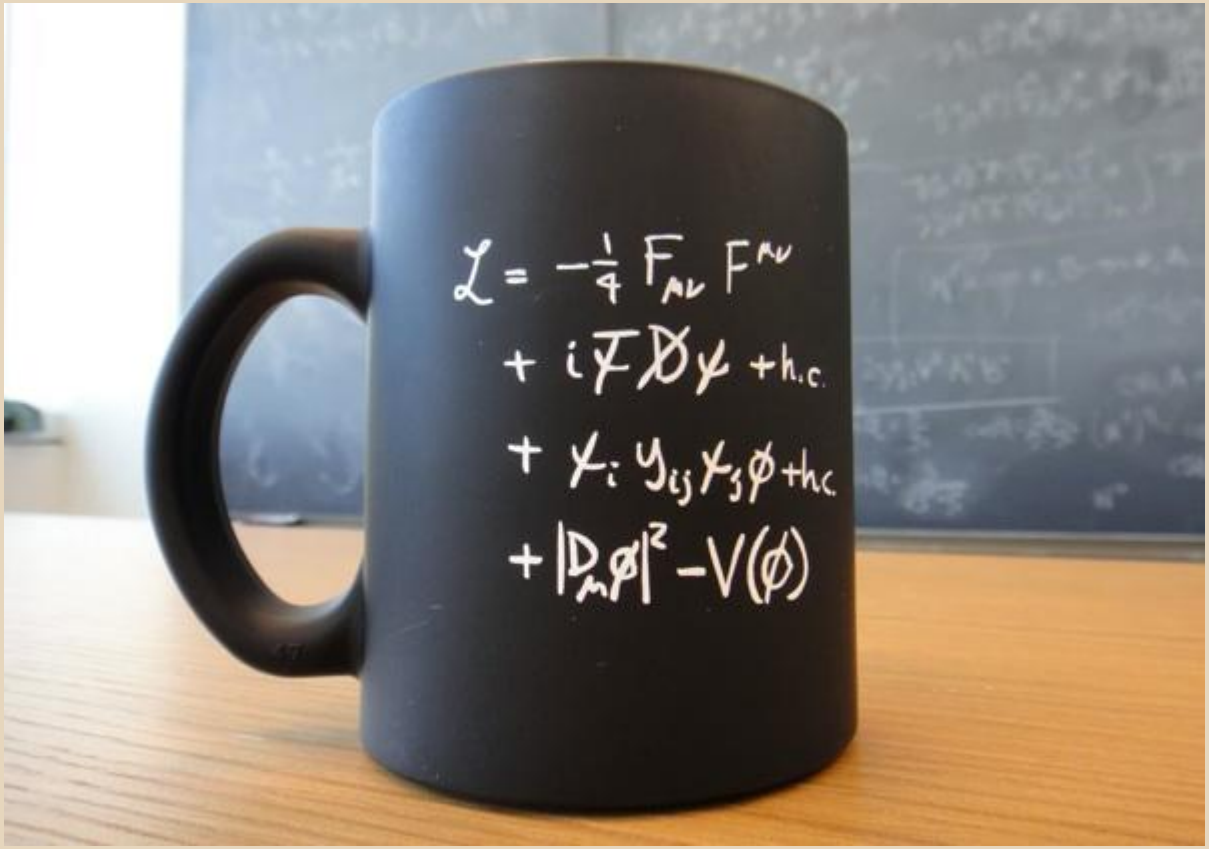
Outline

- The interplay between codes
- LanHep
- MicrOmegas
- Hands-on and homeworks

LanHEP and MicrOmegas



LanHEP



Code for processing a particle physics model i.e. the lagrangian

Install and run

Download

LanHEP version 3.1.7 (updated on February 27, 2013) source

Manual

arxiv:0805.0555

Installation

```
tar xvzf lanhep317.tgz
cd lanhep317
(g)make -> lhep is created
make clean
```

Execution

`lhep` ***`yourmodel.mdl`*** `-ca` (generate CalcHEP files)

`lhep` ***`yourmodel.mdl`*** `-tex` (generate TeX files)

Example Model

1st example: Scalar QED

$$\mathcal{L} = -(D^\mu\varphi)^\dagger D_\mu\varphi - m^2\varphi^\dagger\varphi - \tfrac{1}{4}\lambda(\varphi^\dagger\varphi)^2 - \tfrac{1}{4}F^{\mu\nu}F_{\mu\nu}$$

$$D_\mu \equiv \partial_\mu - ieA_\mu$$

->Scalar_QED.mdl

```
scalar 'phi'/'Phi': (phi_field, mass 'm_phi'=1).
vector A/A:(photon).
let F^mu^nu = deriv^mu*A^nu-deriv^mu*A^nu.
lterm -1/4*(F^mu^nu)**2-1/2(deriv^mu*A^mu)**2.
let Dd^mu=deriv^mu - i*ee*A^mu.
let DD^mu=deriv^mu - i*ee*A^mu.
lterm DD^mu*phi*Dd^mu*Phi.
```

Example Model

1st example: Scalar QED

$$\mathcal{L} = -(D^\mu\varphi)^\dagger D_\mu\varphi - m^2\varphi^\dagger\varphi - \tfrac{1}{4}\lambda(\varphi^\dagger\varphi)^2 - \tfrac{1}{4}F^{\mu\nu}F_{\mu\nu}$$

$$D_\mu \equiv \partial_\mu - ieA_\mu$$

->Scalar_QED.mdl

'particle_name' / 'antiparticle_name'

parameter of the lagrangian

```
scalar 'phi'/'Phi': (phi_field, mass 'm_phi'=1).
vector A/A:(photon).
let F^mu^nu = deriv^mu*A^nu-deriv^mu*A^nu.
lterm -1/4*(F^mu^nu)**2-1/2(deriv^mu*A^mu)**2.
let Dd^mu=deriv^mu - i*ee*A^mu.
let DD^mu=deriv^mu + i*ee*A^mu.
lterm DD^mu*phi*Dd^mu*Phi.
```

Tensor element definition

term of the lagrangian

End of command with a dot!!!

Example model

2nd example: QED

$$\mathcal{L} = \bar{\psi}(i\gamma^\mu D_\mu - m)\psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu}$$

$$\mathcal{L}_{GF} = -\frac{1}{2}(\partial_\mu A^\mu)^2$$

->QED.mdl

```
spinor e1/E1: (electron, mass me=0.0051).
vector A/A:(photon).
let F^mu^nu=deriv^mu*A^nu-deriv^mu*A^nu.
lterm -1/4*(F^mu^nu)**2-1/2(deriv^mu*A^mu)**2.
lterm E1*(i*gamma*deriv+me)*e1.
lterm ee*E1*gamma*A*e1.
```

Example model

2nd example: QED

$$\mathcal{L} = \bar{\psi}(i\gamma^\mu D_\mu - m)\psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu}$$

$$\mathcal{L}_{GF} = -\frac{1}{2}(\partial_\mu A^\mu)^2$$

->QED.mdl

```
spinor e1/E1: (electron, mass me=0.0051).
vector A/A:(photon).
let F^mu^nu=deriv^mu*A^nu-deriv^mu*A^nu.
lterm -1/4*(F^mu^nu)**2-1/2(deriv^mu*A^mu)**2.
lterm E1*(i*gamma*deriv+me)*e1.
lterm ee*E1*gamma*A*e1.
```

gauge coupling

gamma matrix four-vector with lorentz & dirac index already contracted

*in example: lterm ee*E1^a*gamma^a^b^mu*A^mu*e1^b*

Example model

3rd example: QCD

$$L_{YM} = -\frac{1}{4} F^{a\mu\nu} F^{a}_{\mu\nu},$$

$$F^{a}_{\mu\nu} = \partial_\mu G^{a}_{\nu} - \partial_\nu G^{a}_{\mu} - g_s f^{abc} G^{b}_{\mu} G^{c}_{\nu},$$

$$L_F = \bar{q}_i \gamma^\mu \partial_\mu q_i + g_s \lambda^{a}_{ij} \bar{q}_i \gamma^\mu q_j G^{c}_{\mu},$$

$$-\frac{1}{2}(\partial_\mu G^{\mu}_{a})^2 + i g_s f^{abc} \bar{c}^{a} G^{b}_{\mu} \partial^\mu c^{c},$$

->QCD.mdl

```
model QCD/2.
parameter gg=1.117:'Strong coupling'.
spinor q/Q: (quark, mass mq=0.01, color c3).
vector G/G: (gluon, color c8, gauge).
let F^mu^nu^a=deriv^nu*G^mu^a-deriv^mu*G^nu^a-gg*f_SU3^a^b^c*G^mu^b*G^nu^c.
lterm -F**2/4-(deriv*G)**2/2.
lterm Q*(i*gamma*deriv+mq)*q.
lterm i*gg*f_SU3*ccghost(G)*G*deriv*ghost(G).
lterm gg*Q*gamma*lambda*G*q.
```

Example model

3rd example: QCD

$$L_{YM} = -\frac{1}{4} F^{a\mu\nu} F^{a}_{\mu\nu},$$

$$F^{a}_{\mu\nu} = \partial_\mu G^{a}_{\nu} - \partial_\nu G^{a}_{\mu} - g_s f^{abc} G^{b}_{\mu} G^{c}_{\nu},$$

$$L_F = \bar{q}_i \gamma^\mu \partial_\mu q_i + g_s \lambda^{a}_{ij} \bar{q}_i \gamma^\mu q_j G^{c}_{\mu},$$

$$-\frac{1}{2}(\partial_\mu G^{\mu}_{a})^2 + i g_s f^{abc} \bar{c}^{a} G^{b}_{\mu} \partial^\mu c^{c},$$

strong constant definition

color index

->QCD.mdl

```
model QCD/2.
parameter gg=1.117:'Strong coupling'.
spinor q/Q: (quark, mass mq=0.01, color c3).
vector G/G: (gluon, color c8, gauge).
let F^mu^nu^a=deriv^nu*G^mu^a-deriv^mu*G^nu^a-gg*f_SU3^a^b^c*G^mu^b*G^nu^c.
lterm -F**2/4-(deriv*G)**2/2.
lterm Q*(i*gamma*deriv+mq)*q.
lterm i*gg*f_SU3*ccghost(G)*G*deriv*ghost(G).
lterm gg*Q*gamma*lambda*G*q.
```

structure constant of SU(3) group

adjoint representation of the color group

The Standard Model

LanHEP contains several ready-to-use models

The most useful is the model file of the SM

```
workshop_sm.tar.gz
```

uncompress it and execute lanhep

```
./lhep sm.mdl -ca
```

The Standard Model

open `sm.mdl` and fix the line with

```
use standard_model_def.
```

to include the file `standard_model_def.mdl`

`lgrng1.mdl` contains the vertex

`func1.mdl` has the functions defined in the model

`prtcls1.mdl` the particle definition

`var1.mdl` the variables

Your model file

```
keys gauge_fixing=Feynman.

do_if gauge_fixing==Feynman.
    model 'your model name (Feyn. gauge)'/1.
do_else_if gauge_fixing==unitary.
    model 'your model name (un. gauge)'/2.
do_else.
    write('Error: the key "gauge" should be either "Feynman" or "unitary".').
    quit.
end_if.
```

add your model file here

```
CheckHerm.
CheckMasses.
```

Inert (singlet) dark matter

It is the most minimal model

Contains an extra scalar field *S* and the interaction with the *SM* is via the Scalar potential

$$V_{\text{scalar}} = a_2(H^\dagger H - \frac{v^2}{2})S^\dagger S + a_4(S^\dagger S)^2$$

Inert (singlet) dark matter

let's write the model file ....

```
keys gauge_fixing=Feynman.

do_if gauge_fixing==Feynman.
    model 'workshop (Feyn. gauge)'/1.
do_else_if gauge_fixing==unitary.
    model 'workshop (un. gauge)'/2.
do_else.
    write('Error: the key "gauge" should be either "Feynman" or "unitary".').
    quit.
end_if.

use a base_model.<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
% add model definitions <<<<<<<<<<<<<<<<<<<<<<<<<<<<

CheckHerm.
CheckMasses.
```

Inert (singlet) dark matter

```
% the definitions
parameter MDM = 100.0 : 'DM Mass'.
parameter a4DM = 0.1 : 'S^4 couppling',
          a2DM = 0.1 : 'S^2 H^2 couppling'.

scalar '~S'/'~S':('Dark Matter', mass MDM).
prtcproperty pdg:('~S'=10001).

% DM lagrangian
lterm ((deriv*'~S')*deriv*'~S' - MDM**2*'~S'**2)/2.

% scalar potential
lterm -a4DM/4*('~S')**4 - a2DM*(pp*PP-vev2**2/2)*('~S')**2.
```

Inert (singlet) dark matter

```
run your model!

open output files!
```

MicrOmegas

Code to calculate properties of cold dark matter (WIMP) in a generic model of particle physics

for example:

- Relic abundance: solving Boltzmann equation of Early Universe evolution
- Indirect detection cross section i.e. annihilation cross section
- Direct detection cross section: spin independent, spin dependent

The website

lapth.cnrs.fr/micromegas

Current version

http://lapth.cnrs.fr/micromegas/downloadarea/code/micromegas_2.4.5.tgz

Install and run

Download latest version (2.4.5) from http://lapth.cnrs.fr/micromegas/

Unpack the tarfile

```
tar xzvf micromegas_2.4.5.tgz
```

Compile MicrOmegas

```
cd micromegas_2.4.5
(g)make
```

Create a project (called workshop)

```
./newProject workshop
cd workshop
```

Adding your model

Copy **`*1.mdl`** files previously generated with LanHEP in the folder:

`workshop/work/models`

Edit the file

`main.c` (you can rename it)

compile

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

and run

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

the default `main.c`

```
#define MASSES_INFO
#define OMEGA
#define INDIRECT_DETECTION

/*#define RESET_FORMFACTORS*/
  /* Modify default nucleus form factors,
    DM velocity distribution,
    A-dependence of Fermi-dencity
  */
#define CDM_NUCLEON
```

There are switches for calculating some observable with values defined in your lanhep modelfile. Comment // to disable

modifying the values

```
// add inside main()
int foo;
double MDM2, double a2DM2, double a4DM2;

MDM2 = 100.0;
a2DM2 = 0.1;
a4DM2 = 0.1;

foo=assignVal("MDM", MDM2);
foo=assignVal("a4DM", a4DM2);
foo=assignVal("a2DM", a2DM2);
```

MDM, a4DM, a2DM are the names that apear on the LanHEP file vars1.mdl

Obtaining the relic abundance

```
#ifdef OMEGA
{ int fast=1;
  double Beps=1.E-5, cut=0.01;
  double Omega,Xf;
  printf("\n==== Calculation of relic density =====\n");
  Omega=darkOmega(&Xf,fast,Beps);
  printf("Xf=%.2e Omega=%.2e\n",Xf,Omega);
  printChannels(Xf,cut,Beps,1,stdout);
}
#endif
```

the value correspond to Ωh^2 which is a constrain!!!

annihilation cross section

```
#ifdef INDIRECT_DETECTION
{
...
  printf("\n==== Indirect detection =======\n");

  sigmaV=calcSpectrum(4,SpA,SpE,SpP,SpNe,SpNm,SpNl ,&err);
    /* Returns sigma*v in cm^3/sec.     SpX - calculated spectra of annihilation.
       Use SpectdNdE(E, SpX) to calculate energy distribution in  1/GeV units.

       First parameter 1-includes W/Z polarization
                       2-includes gammas for 2->2+gamma
                       4-print cross sections
    */
  printf("sigmav=%.2E[cm^3/s]\n",sigmaV);
...
}
```

Direct detection

```
#ifdef CDM_NUCLEON
{ double pA0[2],pA5[2],nA0[2],nA5[2];
  double Nmass=0.939; /*nucleon mass*/
  double SCcoeff;
  printf("\n==== Calculation of CDM-nucleons amplitudes  =====\n");
  nucleonAmplitudes(NULL, pA0,pA5,nA0,nA5);
  printf("CDM[antiCDM]-nucleon micrOMEGAs amplitudes:\n");
  printf("proton:  SI  %.3E [%.3E]  SD  %.3E [%.3E]\n",pA0[0], pA0[1],  pA5[0], pA5[1]
);
  printf("neutron: SI  %.3E [%.3E]  SD  %.3E [%.3E]\n",nA0[0], nA0[1],  nA5[0], nA5[1]
);
  ...
}
#endif
```

Hands-on

Modify the code to scan in the space of parameter:

MDM, a2DM, a4DM

10 GeV < MDM < 500 GeV

-1 < a2DM < 1

a4DM = 1

and do plots *annihilation cross section* vs *mass*
impose the constraint

0.98*0.11 < Ωh² < 1.02*0.11

Higgs decay

```
void width_higgs(double *width){
    int i;
    txtList LL;
    int dim;
    *width = 0.0;
    for(i=0;i<nModelParticles;i++){
        if (!strcmp(ModelPrtcls[i].name,"H")) {
            *width += pWidth(ModelPrtcls[i].name, &LL,&dim);
            //printf("H %.5E \n", *width);
        }
    }
}
```

Invisible higgs decay

```
chk=assignVal("MDM", *MDM);
chk=assignVal("b4DM", 0.0);
chk=assignVal("a2DM", 0.0);
chk=assignVal("gDM", 0.0);
ForceUG=0;  /* to Force Unitary Gauge assign 1 */

chk=sortOddParticles(cdmName);
if(chk) {printf("Can't calculate %s\n",cdmName); return 1;}

double whSM;
width_higgs(&whSM); // calculating the SM width
```

Invisible higgs decay

```
chk=assignVal("b4DM", *b4DM);
chk=assignVal("a2DM", *a2DM);
chk=assignVal("gDM", *gDM);

chk=sortOddParticles(cdmName);
if(chk) { printf("Can't calculate %s\n",cdmName); return 1;}

double whTOT;
width_higgs(&whTOT);

*invhiggs = (whTOT-whSM)/whTOT;
if (*invhiggs > BR_INV_HIGGS_HI) { err = 1; } // bounds from LHC
```

Homework

inert double dm model (use: 1003.3125)

$$V = \mu_1^2|H_1|^2 + \mu_2^2|H_2^2| + \lambda_1|H_1|^4 + \lambda_2|H_2|^4 + \lambda_3|H_1|^2|H_2|^2 + \lambda_4|H_1^\dagger H_2|^2 + \frac{\lambda_5}{2}\left[(H_1^\dagger H_2)^2 + \text{h.c.}\right],$$

$$m_{H^\pm}^2 = \mu_2^2 + \frac{1}{2}\lambda_3, v^2,$$

$$m_{H^0}^2 = \mu_2^2 + \frac{1}{2}(\lambda_3 + \lambda_4 + \lambda_5)v^2,$$

$$m_{A^0}^2 = \mu_2^2 + \frac{1}{2}(\lambda_3 + \lambda_4 - \lambda_5)v^2\,,$$

Thank you