Hands-on LanHEP and CalcHEP: Efficient Implementation and Exploration of Dark Matter Models

Alexander Belyaev

a.belyaev@soton.ac.uk
https://launchpad.net/calchep
https://launchpad.net/lanhep

Southampton University & Rutherford Appleton Lab

Dark Tools, Turin, 16-19 June, 2025

LanHEP package

Andrei Semenov **http://theory.sinp.msu.ru/~semenov/lanhep.html**

Comput.Phys.Commun. 201 (2016) 167-170 • e-Print: 1412.5016 [physics.comp-ph]
Comput.Phys.Commun. 180 (2009) 431-454 • e-Print: 0805.0555 [hep-ph]
LanHEP: A Package for automatic generation of Feynman rules in gauge models, hep-ph/9608488

LanHEP is a powerful tool for automatically generating Feynman rules directly from a model's Lagrangian, written in a compact and intuitive form similar to that used in publications.

Key Features

- Writes your Lagrangian in a clean and readable format.
- Automatically obtains Feynman rules in terms of physical fields and independent parameters.
- Supports summation over indices and recognises commonly used expressions such as covariant derivatives and gauge field strength tensors
- Export models in LaTeX, CompHEP, CalcHEP, UFO, and FeynArts formats
- Supports all class of models with particles upto spin 2 making it a key tool for BSM studies
- **Simple and User-Friendly Setup**
  - Requires only a C compiler (e.g. gcc) to build
  - You don't need to know the C language — LanHEP uses its own intuitive input format.

Many advanced features

- Supports superpotential formalism
- Checks: BRST invariance, electric charge conservation, hermiticity
- Supports multiplets, matrices and much more — see the manual and this lecture

LanHEP Installation Guide

1. Download the code

a) **from Dark Tools github**

wget https://raw.githubusercontent.com/dimauromattia/darktools/main/lanhep/lhep403.tgz

or

b) **from the HEP Tools site** → Link to HEP TOOLS directory → lanhep → lhep403.tgz

2. Unpack the archive

tar -zxvf lhep403.tgz

3. Enter the directory

cd lanhep403

4. Compile

make

The **lhep** *executable will be built — you can add it to your PATH for easy use, e.g. add*

export PATH="$PATH:/path-to-lhep-file/" line to .bashrc

ex#1 *install LanHEP*

Running LanHEP

- **example of LanHEP running**

```
cd mdl
../lhep -ca stand.mdl
```

File sm_tex processed, 0 sec.
File stand.mdl processed, 0 sec.

where stand.mdl is the model in LanHEP format

- **running lhep without arguments** gives you the list of options for output formats

```
../lhep
Error: output format is not set. Use command line options:
        -CompHEP or -co for CompHEP format;
        -CalcHEP or -ca for CalcHEP format;
        -FeynArts or -fa for FeynArts format;
        -ufo for UFO format;
        -tex for LaTeX format.
```

- e.g. you can run

```
../lhep -ufo stand.mdl
../lhep -tex stand.mdl
```

to produce model in UFO format and the LaTeX formats respectively

- `../lhep -ca -OutDir outdir stand.mdl`

to write results into **`outdir`**

An example of simplest model - QED

$$\mathcal{L}_{QED} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{e}\gamma^{\mu}(i\partial_{\mu} + g_e A_{\mu})e - m\bar{e}e, \qquad \mathcal{L}_{GF} = -\frac{1}{2}(\partial_{\mu}A^{\mu})^2$$

```
model QED/1.
parameter ge=0.31333: 'Electric charge'.
spinor e1/E1:(electron, mass me=0.000511).
vector A/A:(photon).
let F^mu^nu=deriv^nu*A^mu-deriv^mu*A^nu.
lterm -1/4*(F^mu^nu)**2 - 1/2*(deriv^mu*A^mu)**2.
lterm E1*(i*gamma*deriv+ge*gamma*A-me)*e1.
```

qed.mdl

```
../lhep -ca qed.mdl
```

will produce model files in CalcHEP format

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

```
../lhep -tex qed.mdl
```

will produce model files in LaTeX format

prtcls1.tex, lgrng1.tex, vars1.tex, func1.tex

lgrng1.tex contains

Fields in the vertex	Variational derivative of Lagrangian by fields
$E1_a \quad e1_b \quad A_\mu$	$ge \cdot \gamma^{\mu}_{ab}$

The model structure in CalcHEP format: a set of ASCII tables

`prtcls1.mdl` : the list of particles and their properties

```
QED
 Particles
Full   Name     | P | aP|  number  |spin2|mass|width|color|aux| LaTeX(A)       |  LateX(A+)        |
electron        |e1 |E1 |11        |1    |me  |0    |1    |   |e1              |E1                 |
photon          |A  |A  |22        |2    |0   |0    |1    |   |A               |A                  |
```

`lgrng1.mdl` : particles interactions

```
QED
 Lagrangian
P1   |P2   |P3   |P4   |>   Factor  <|> dLagrangian/ dA(p1) dA(p2) dA(p3)     <|
E1   |e1   |A    |     |ge           |G(m3)
```

`vars1.mdl` : independent model parameters

```
QED
 Variables
 Name   | Value       |>  Comment                                  <|
ge      |0.31333      |Electric charge
me      |0.000511     |mass of electron
```

other two files func xxx.mdl (dependent parameters) and extlib xxx.mdl (libraries to be linked) -- are empty for this simple model

Syntax of LanHEP

- The LanHEP input file is the sequence of statements, each starts with a **special identifier** (such as parameter, lterm, etc) and **ends with the full-stop '.'** symbol. Statement can occupy several lines
- **Identifiers:** Identifiers are the names of particles, parameters etc.
- **Constants:** integers, floating point numbers, strings
- **Comments:** '%' ,'/*' ... '*/'
- **Order of the indices of the objects** (default, can be changed): `[spinor, color c3, color c8, vector]`

- declaring new groups:

```
group color:SU(3).
repres color:(c3/c3b,c8).
```

- parameters `parameter` *name=value:comment.*

- particles

`scalar` *P/aP*`:(`*options*`).`
`spinor` *P/aP*`:(`*options*`).`
`vector` *P/aP*`:(`*options*`).`

See manual and this lecture for more details

You can see/update default options are in calchep.rc file

```
% Definitions specific for CalcHEP format of Feynman rules.

keys OutputFormat=CalcHEP.

prtcformat fullname: 'Full   Name     ',
           name:' P ', aname:' aP', pdg:'  number  ',
           spin2,mass,width, color, aux, texname: ' LaTeX(A)       ',
           atexname:'  LateX(A+)         '.

prtcproperty pdg:(A=22, Z=23, 'W+'=24, G=21,
                 q=1,
                 d=1, u=2, s=3, c=4, b=5, t=6,
                 n1=12, n2=14, n3=16,
                 e1=11, e2=13, e3=15,
                 ne=12, nm=14, nl=16,
                 e=11, m=13, l=15,
  ~ne=1000012, ~nm=1000014, ~nl=1000016,
  ~e1=1000011, ~m1=1000013, ~l1=1000015,
  ~e2=2000011, ~m2=2000013, ~l2=2000015,
  ~eL=1000011, ~mL=1000013,
  ~eR=2000011, ~mR=2000013,

  ~u1=1000002, ~c1=1000004, ~t1=1000006,
  ~u2=2000002, ~c2=2000004, ~t2=2000006,
  ~uL=1000002, ~cL=1000004, ...
```

Next example - QCD

- **Gauge interactions** $L_{YM} = -\frac{1}{4}F^{a\mu\nu}F^{a}_{\mu\nu}, F^{a}_{\mu\nu} = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu - g_s f^{abc} G^b_\mu G^c_\nu,\ G^a_\mu(x)$
- **Quark kinetic term** $L_F = \bar{q}_i \gamma^\mu \partial_\mu q_i + g_s \lambda^a_{ij} \bar{q}_i \gamma^\mu q_j G^c_\mu,$
- **Gauge fixing term and Fadeev-Popov ghost term**

$$\mathcal{L}_{GF} = -\frac{1}{2}(\partial_\mu G^\mu_a)^2 + i g_s f^{abc} \bar{c}^a G^b_\mu \partial^\mu c^c,$$

- **LanHEP model file (qcd.mdl):**

```
model      QCD/2.
parameter  gg= 1.13 : 'Strong coupling'.
vector     G/G: (gluon, color c8, gauge).
spinor     q:(quark, color c3, mass Mq=0.02).
lterm      i*gg*f_SU3*ccghost(G)*G*deriv*ghost(G).
lterm      Q*gamma*(i*deriv + gg*lambda*G)*q.
lterm      -F**2/4  where
              F=deriv^mu*G^nu^a-deriv^nu*G^mu^a+
              i*gg*f_SU3^a^b^c*G^mu^b*G^nu^c.
```

```
../lhep -tex qcd.mdl
```

will produce: vars2.tex, prtcls2.tex, lgrng2.tex

lgrng2.tex

Fields in the vertex	Variational derivative of Lagrangian by fields
$G_{\mu p}\quad G_{\nu q}\quad G_{\rho r}$	$gg f_{pqr}(p_3^\nu g^{\mu\rho} - p_3^\mu g^{\nu\rho} + p_1^\rho g^{\mu\nu} - p_1^\nu g^{\mu\rho} - p_2^\rho g^{\mu\nu} + p_2^\mu g^{\nu\rho})$
$G.C_p\quad G.c_q\quad G_{\mu r}$	$-gg \cdot p_2^\mu f_{pqr}$
$Q_{ap}\quad q_{bq}\quad G_{\mu r}$	$gg \cdot \lambda^r_{pq}\gamma^\mu_{ab}$
$G_{\mu p}\quad G_{\nu q}\quad G_{\rho r}\quad G_{\sigma s}$	$gg^2(g^{\mu\rho}g^{\nu\sigma}f_{pqt}f_{rst} - g^{\mu\sigma}g^{\nu\rho}f_{pqt}f_{rst} + g^{\mu\nu}g^{\rho\sigma}f_{prt}f_{qst}$ $-g^{\mu\sigma}g^{\nu\rho}f_{prt}f_{qst} + g^{\mu\nu}g^{\rho\sigma}f_{pst}f_{qrt} - g^{\mu\rho}g^{\nu\sigma}f_{pst}f_{qrt})$

Vertices with colour particles in CalcHEP

- The CalcHEP Lagrangian tables do not describe explicitly the colour structure of a vertex.
- If colour particles are present in the vertex, the following implicit contractions are assumed (p, q, r are colour indices):

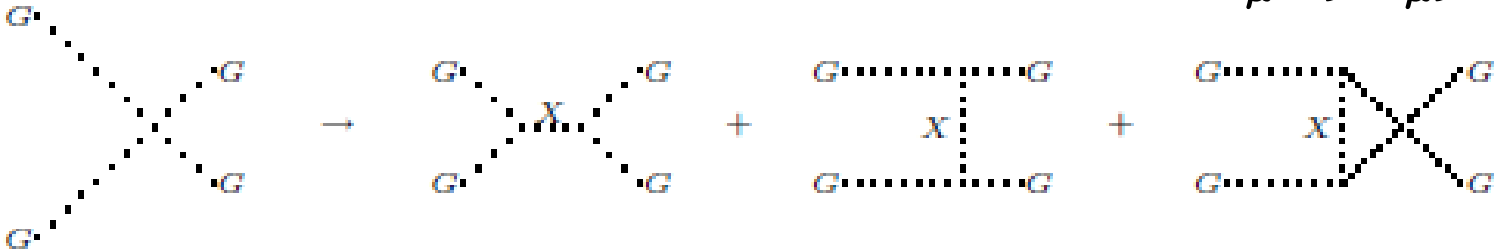
δ_{pq} for two color particles A^1_p and A^2_q

λ^r_{pq} for three particles, which are colour triplet, antitriplet and octet

f^{pqr} for three colour octets $\quad f^{pqr} G^p_\mu G^q_\nu G^r_\lambda$

Vertices with coloure particles in CalcHEP

- *4-gluon vertex can be split it into 3-legs vertices* $f^{pqr} G^q_\mu G^r_\nu X^p_{\mu\nu}$



- *Here the field* $X^p_{\mu\nu}$ *is a Lorenz tensor and colour octet, and this field has constant propagator*
- *If gluon name in CalcHEP is 'G'*
- *the name 'G.t' is used for this tensor particle; its indices are denoted as 'm' and 'M'*

```
QCD
 Lagrangian
P1   |P2   |P3   |P4   |>   Factor <|> dLagrangian/ dA(p1) dA(p2) dA(p3)                                                    <|
G    |G    |G    |     |gg          |m2.p3*m1.m3-m1.p3*m2.m3+m3.p1*m1.m2-m2.p1*m1.m3-m3.p2*m1.m2+m1.p2*m2.m3
G.C  |G.c  |G    |     |-gg         |m3.p2
Q    |q    |G    |     |gg          |G(m3)
G    |G    |G.t  |     |gg*Sqrt2/2  |m1.m3*m2.M3-m1.M3*m2.m3
```

`m1.m2` in CalcHEP etc means g^{m1m2}

Vertices with colour particles in LanHEP

- The splitting of vertex with 4 colour particle into 3-particles vertices is done by LanHEP automatically: each vertex containing 4 colour particles is split to 2 vertices which are joined by automatically generated auxiliary field
- option SplitCol1=N.

  where N is a number:
  - *-1 remove all vertices with 4 color particles from Lagrangian;*
  - *0 turn off multiplet level vertices splitting;*
  - *1 allows vertices splitting with 4 color multiplets;*
  - *2 allows vertices splitting with any 4 scalar multiplets except Higgs* **[default]**
- *option SplitCol2=N.*
  - *where N is a number:*
  - *0 disable vertex level splitting;*
  - *1 enable vertex level splitting (only for vertices with 4 color particles)* **[default]**

More details on LanHEP options and syntax

- ***Specials*** `gamma`, `gamma5`, `moment`, `deriv`, `lambda`, `f_SU3`
  declaring new specials: `special` *name*`:(`*islist*`).`
- ***Orthogonal matrices*** `OrthMatrix( {{`a_{11}`,` a_{12}`}, {`a_{21}`,` a_{22}`}} ).`
- ***Including files*** `read` *file*`.` or `use` *file*`.` (no multiple reading)
- ***Checking electric charge conservation*** `SetEM(`*photon*`,` *param*`).`
- ***Running LanHEP*** `lhep` *filename options*

  `-OutDir` *directory* Set the directory for output files

  `-InDir` *directory* Set the directory where to search files

  `-tex` LanHEP generates LaTeX files

  `-frc` If `-tex` option is set, forces LanHEP to split 4-fermion and 4-color vertices just as it is made for CompHEP files.

  `-texLines` *num* Set number of lines in LaTeX tables

  `-texLineLength` *num* Controls width of the Lagrangian

Default groups and specials in LanHEP: mdl/lhep.rc

```
special gamma:(spinor,cspinor,vector).
special gamma5:(spinor,cspinor).
special '(1+gamma5)/2':(spinor,cspinor), '(1-gamma5)/2':(spinor,cspinor).
special moment:(vector).
special '__moment__start__':(vector), '__moment__end__'.
special epsv:(vector,vector,vector,vector).
special eps3v:(vector,vector,vector).
group color:SU(3).
repres color:(c3/c3b,c8,c6/c6b).
SetDefIndex(spinor,color c3, color c8, vector).
special lambda:(color c3, color c3b, color c8).
special f_SU3:(color c8, color c8, color c8).
special d_SU3:(color c8, color c8, color c8).
special eps_c3: (color c3,  color c3,  color c3),
        eps_c3b:(color c3b, color c3b, color c3b).
special k_c6:(color c3b, color c3b, color c6),
        k_c6b:(color c3, color c3, color c6b).
special l_c6:(color c6, color c6b, color c8).
let deriv=-i*moment.
special deriv5.
let '__deriv__start__'=-i*'__moment__start__'.
let tau1={{0,1},{1,0}}, tau2={{0,i},{-i,0}}, tau3={{1,0},{0,-1}}.
let tau={tau1,tau2,tau3}.
```

SM Lagrangian with LanHEP

The latest version of SM implementation comes with CalcHEP and micrOMEGAs, located in **model_src** folder of the CalcHEP distribution (also in LanHEP mdl folder)
model_src contains LanHEP source files for SM and Inert Doublet Model

```
model_src: sm.inc idm.lhep sm.lhep README
```

sm.lhep

```
keys gauge_fixing=Feynman, CKMdim=1, hgg=On,
h4G=Off.
do_if hgg==On.
    do_if h4G==On.
        model  'SM(+hgg+h4G)'/3.
    do_else.
        model 'SM(+hgg)'/2.
    end_if.
do_else.
    model 'SM'/1.
end_if.

read 'sm.inc'.
CheckHerm.
CheckMasses.
```

SM Lagrangian with LanHEP: sm.inc - parameters

```
% Standard Model - unitary and t'Hooft-Feynman gauges.
external_func(alphaQCD,1).
external_func(McEff,1).
. . . . . .
parameter  EE  = 0.31333 : 'Electromagnetic coupling constant (<->1/128)',
       GG  = 1.117   : 'Strong coupling constant (Z point)  (PDG-94)',
       SW  = 0.4740  : 'sin of the Weinberg angle 0.474 - "on-shell",481 - "MS-bar" )'.

parameter  CW  = sqrt(1-SW**2) : 'cos of the Weinberg angle'.

parameter Q=100: 'Scale of effective running masses',
           MW=80.385: 'W boson mass',
           GF=EE**2/(2*SW*MW)**2/Sqrt2 :' experimental value 1.166E-5 [1/GeV^2]',
           MZ=MW/CW:  'Z boson mass',
           Mtp  =172.5:'Top quark pole mass',
           McMc =1.23:     'Mc(Mc)  MS-BAR',
           MbMb =4.25:     'Mb(Mb)  MS-BAR',
           alphaSMZ=0.1184:'Srtong alpha(MZ)',
           LamQCD=initQCD5(alphaSMZ,McMc,MbMb,Mtp),
           Mb=MbEff(Q),
           Mc=McEff(Q),
           Ms=MqEff(0.096,Q): 's-quark effective mass  via 2MeV running one' .
```

SM Lagrangian with LanHEP: sm.inc - particles

```
do_if gauge_fixing==Feynman.
vector
    A/A: (photon, gauge),
    Z/Z:('Z boson', mass MZ, width wZ = auto, gauge),
    G/G: (gluon, color c8, gauge),
        'W+'/'W-': ('W boson', mass MW, width wW = auto, gauge).
do_else.

vector
    A/A: (photon, gauge),
    Z/Z:('Z boson', mass MZ, width wZ = auto),
    G/G: (gluon, color c8, gauge),
        'W+'/'W-': ('W boson', mass MW, width wW = auto).
end_if.

spinor  ne/Ne:(neutrino,left),        e:(electron),
        nm/Nm:('mu-neutrino',left),   m:(muon),
        nl/Nl:('tau-neutrino',left),  l:('tau-lepton', mass Ml  = 1.777).

spinor  u:('u-quark',color c3),
        d:('d-quark',color c3),
        c:('c-quark',color c3, mass Mc),
        s:('s-quark',color c3, mass Ms), . . .
```

SM Lagrangian with LanHEP: sm.inc – CKM,'lets'

```
parameter    s23 = 0.040   : 'Parameter of C-K-M matrix (PDG-94)',
             s13 = 0.0035  : 'Parameter of C-K-M matrix (PDG-94)',
             c23  = sqrt(1-s23**2),
             c13  = sqrt(1-s13**2).

parameter  Vud = c12*c13                  ,
           Vus = s12*c13                  ,
           Vub = s13                 ,
           Vcd = (-s12*c23-c12*s23*s13) ,
           Vcs = (c12*c23-s12*s23*s13)  ,
           Vcb = s23*c13                  ,
           Vtd = (s12*s23-c12*c23*s13)    ,
           Vts = (-c12*s23-s12*c23*s13)   ,
           Vtb = c23*c13                .
OrthMatrix( { {Vud,Vus,Vub}, {Vcd,Vcs,Vcb}, {Vtd,Vts,Vtb}} ).
. . . . .
let l1={ne,e}, L1={Ne,E}.
let l2={nm,m}, L2={Nm,M}.
let l3={nl,l}, L3={Nl,L}.

let q1={u,d}, Q1={U,D}, q1a={u,Vud*d+Vus*s+Vub*b}, Q1a={U,Vud*D+Vus*S+Vub*B}.
let q2={c,s}, Q2={C,S}, q2a={c,Vcd*d+Vcs*s+Vcb*b}, Q2a={C,Vcd*D+Vcs*S+Vcb*B}.
let q3={t,b}, Q3={T,B}, q3a={t,Vtd*d+Vts*s+Vtb*b}, Q3a={T,Vtd*D+Vts*S+Vtb*B}.
```

SM Lagrangian with LanHEP: sm.inc – interactions

```
%================== Self-interaction of gauge bosons

lterm -F**2/4  where  F=deriv^mu*B1^nu-deriv^nu*B1^mu.
lterm -F**2/4  where  F=deriv^mu*G^nu^a-
deriv^nu*G^mu^a+i*GG*f_SU3^a^b^c*G^mu^b*G^nu^c.
lterm -F**2/4  where  F=deriv^mu*W^nu^a-deriv^nu*W^mu^a+g2*eps^a^b^c*W^mu^b*W^nu^c.


%================== left fermion interaction with gauge fields

lterm   i*anti(psi)*gamma*(1-g5)/2*(deriv-i*g2*tau*W/2-i*Y*g1*B1)*psi
        where
            psi=l1,  Y=-1/2;
            psi=l2,  Y=-1/2;
            psi=l3,  Y=-1/2;
            psi=q1a, Y= 1/6;
            psi=q2a, Y= 1/6;
            psi=q3a, Y= 1/6.
```

$H\gamma\gamma$ effective vertex with LanHEP

- The `sm.inc` contains the $H\gamma\gamma$ implementation through $\phi_H^2 F^{\mu\nu} F_{\mu\nu}$

```
let shd = {  i*'W+.f',  (vev(vevh)+h-i*'Z.f')/Sqrt2 }.
external_func(lAAhiggs,2).

parameter LAAh=-cabs(lAAhiggs(Mh,str(h))).
 lterm  LAAh*(shd*anti(shd)-vevh**2/2)/vevh*F**2  where
              F=deriv^mu*A^nu-deriv^nu*A^mu.
```

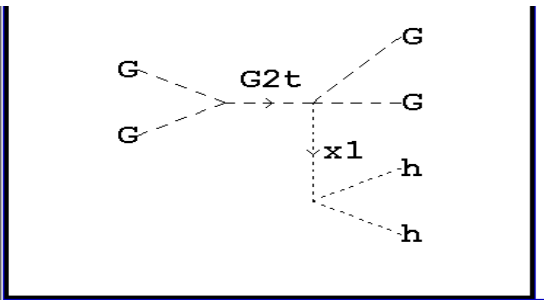
lAAhiggs(Mh,str(h)) is universal function which works with any BSM model – looks at any vertex which would contribute to $H\gamma\gamma$

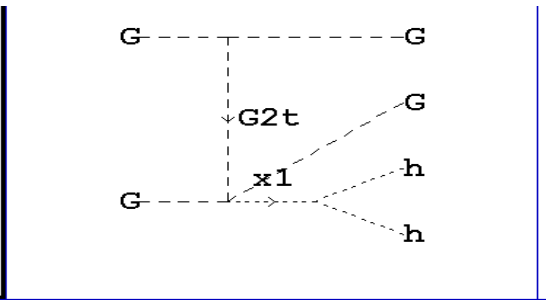
lGGhiggs(Mh,str(h)) is universal function which works with any BSM model – looks at any vertex which would contribute to Hgg

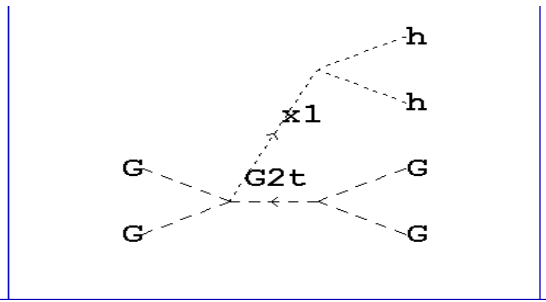
Hgg and auxiliary particles with LanHEP

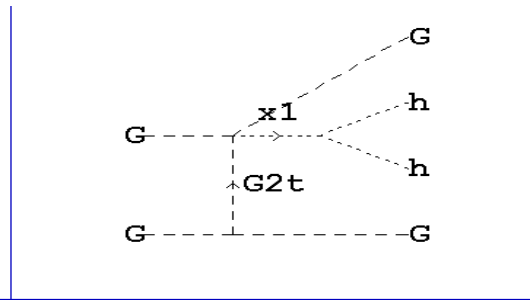
$\phi_H^2 F^{\mu\nu} F_{\mu\nu}$ effective Lagrangian leads to 6-point Hhgggg vertex, LanHEP allows to realise this via auxiliary non-propagating fields

```
vector  G2t/G2T: (G2T, mass Maux, color c8, aux ('!*')).
scalar  x1/X1:   (x1,  mass Maux, aux ("!*")).
parameter  LGGh=-cabs(lGGhiggs(Mh,str(h))).

lterm 1/vevh*LGGh*RQCDh*(shd*anti(shd)-vevh**2/2-vevh*h)*x1*Maux.

lterm (GG/2*f_SU3^a^b^c*G^n^a*G^m^b*'G2t.t'^m^n^c)*(X1*Maux).
lterm    GG/2*f_SU3^a^b^c*G^n^a*G^m^b*'G2T.t'^m^n^c.
```









Inert Higgs Doublet model implementation: `idm.lhe`

Inert Doublet model contains (IDM) two $SU(2)$ doublets

$$H_1 = \begin{pmatrix} 0 \\ \langle v \rangle + h/\sqrt{2} \end{pmatrix}, \quad H_2 = \begin{pmatrix} \tilde{H}^+ \\ (\tilde{X} + i \cdot \tilde{H}_3)/\sqrt{2} \end{pmatrix}$$

The Lagrangian contains only *even* powers of H_2 doublet

$$\begin{aligned} L &= (\text{SM terms}) \\ &+ D^\mu H_2^\dagger D_\mu H_2 - \mu^2 H_2^2 - \lambda_2 H_2^4 - \lambda_3 H_1^2 H_2^2 - \lambda_4 (H_1^\dagger H_2)(H_2^\dagger H_1) \\ &- \frac{\lambda_5}{2}[(H_1^\dagger H_2)^2 + (H_2^\dagger H_1)^2] \end{aligned}$$

Because of symmetry $H_2 \to -H_2$, 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 : $\lambda_2, \qquad \lambda_L = \lambda_3 + \lambda_4 + \lambda_5$

IDM implementation: idm.lhe-parameters

```
keys gauge_fixing=Feynman, CKMdim=1, hgg=Off,h4G=Off.
do_if hgg==On.
    do_if h4G==On.
        model  'IDM(+hgg+h4G)'/5.
    do_else.
        model 'IDM(+hgg)'/5.
    end_if.
do_else.
    model 'IDM'/4.
end_if.

read 'sm.inc'.

parameter MHX=111,MH3=222,MHC=333.
parameter laL=0.01, la2=0.01.
parameter mu2=MHX**2-laL*(2*MW/EE*SW)**2.
parameter la3=2*(MHC**2-mu2)/(2*MW/EE*SW)**2.
parameter la5=(MHX**2-MH3**2)/(2*MW/EE*SW)**2.
parameter la4=2*laL-la3-la5.
```

IDM implementation: – particles & interactions

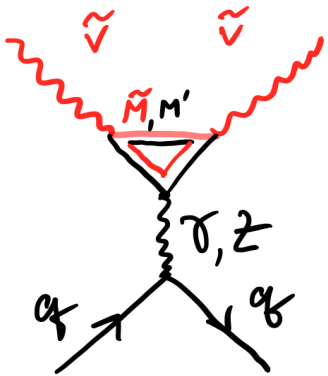
```
scalar '~H3'/'~H3':('odd Higgs',pdg 36, mass MH3, width wH3 = auto).
scalar '~H+'/'~H-':('Charged Higgs',pdg 37, mass MHC, width wHC=auto).
scalar '~X'/'~X'  :('second Higgs', pdg 35,  mass MHX, width wHX=auto).

let sId = { -i*'~H+',  ('~X'+i*'~H3')/Sqrt2 },    % scalar inert doublet
    sID = anti(sId).                              % scalar inert anti-doublet

let DsId^mu^a = (deriv^mu-i*g1/2*B1^mu)*sId^a - i*g2/2*tau^a^b^c*W^mu^c*sId^b.
let DsID^mu^a = (deriv^mu+i*g1/2*B1^mu)*sID^a + i*g2/2*tau^b^a^c*W^mu^c*sID^b.

lterm DsID*DsId.
lterm -mu2*sId*sID.
lterm -la2*(sId*sID)**2.
lterm -la3*(shd*shD)*(sId*sID).
lterm -la4*(shd*sID)*(shD*sId).
lterm -la5/2*(shd*sID)**2 + AddHermConj.

CheckHerm.
CheckMasses.
```

Fermionic Portal for Vector Dark Matter (FPVDM)

- It is the framework, representing the class of models
  [Deandrea, Moretti, Panizzi, Ross, Thongyoi, AB – arXiv:2204.03510,2203.04681]

- Various realisations are possible, including one or several VL fermions

$$\begin{aligned}
\mathcal{L}_{FPVDM} &= -\frac{1}{4}(V^i_{D\mu\nu})^2 + \bar{\Psi} i D \Psi + |D_\mu \Phi_D|^2 - V(\Phi_H, \Phi_D) \\
&- (y'_{\alpha\beta} \bar{\Psi}^{i\alpha}_L \Phi_D f^{\mathrm{SM}\beta}_R + h.c) - {M_\Psi}^{ij} \bar{\Psi}^i \Psi^j \\
V(\Phi_H, \Phi_D) &= -\mu_H^2 \Phi_H^\dagger \Phi_H - \mu_D^2 \Phi_D^\dagger \Phi_D + \lambda_H (\Phi_H^\dagger \Phi_H)^2 \\
&+ \lambda_D (\Phi_D^\dagger \Phi_D)^2 + \lambda_{HD} (\Phi_H^\dagger \Phi_H)(\Phi_D^\dagger \Phi_D)
\end{aligned}$$



- $y'_{\alpha\beta}$ can have a flavour structure – to explain flavour anomalies
- λ_{HD} can be negligible at tree-level, DM can be well-generated via FP
- the model with $\Psi = \begin{pmatrix} \tilde{T} \\ T \end{pmatrix}$ and $\lambda_{HD} = 0$ was explored

FPVDM implementation in LanHEP

- Available at hepmdb: `https://hepmdb.soton.ac.uk/hepmdb:0322.0335`
  together with other many DM models for micrOMEGAs
  See also **m335.xxx.zip** and **s335.xxx.zip** in lanhep forder at DarkToll github

- The LanHEP file is: `FPVDM_with_TOP_final.lhep`

- This model implementation uses various functions from `SLHAplus`
  `Belanger, Christensen, Pukhov, Semenov` CPC 182 (2011) 763-774, 1008.0181 [hep-ph]

```
external_func(rDiagonal,*).
external_func(MassArray,*).
external_func(MixMatrix,*).
```

```
OrthMatrix( {{H11,H21}, {H12,H22}} ).
parameter   Hessian = rDiagonal(2,H11,H12,H22).
parameter   EiHess1 = MassArray(Hessian,1),
            EiHess2 = MassArray(Hessian,2).
```

- It also uses external function from additional library which includes loops to define the form-factors(FF) of DM,DM, photon interactions
  `external_func(FF,12).`

- There is also a command to add a line to extlib file:
  `extlib '$clib/VP.c $clib/vp0.c -I$clib -lm'.`

Using the superpotential formalism in the MSSM and its extensions

- Superpotential – a polynomial W depending on scalar fields A_i
- The most general form of the MSSM superpotential which does not violate gauge invariance and the SM conservation laws is:

$$W = \mu\epsilon_{ij}H_i^1H_j^2 + \epsilon_{ij}Y_l^{IJ}H_i^1L_j^IR^J + \epsilon_{ij}Y_d^{IJ}H_i^1Q_j^ID^J + \epsilon_{ij}Y_u^{IJ}H_i^2Q_j^IU^J$$

which in LanHEP notation will take a form

```
keep_lets W.
let W=eps*(mu*H1*H2+ml*H1*L*R+md*H1*Q*D+mu*H2*Q*U).
```

Where H1, H2, L, R, Q, U, D should be defined above as doublets and singlets in terms of scalar particles.

keep_lets statement substitution of H1, H2, L, R, Q, U, D in terms of their components

Using the superpotential formalism in the MSSM and its extensions

- Yuakawa interactions are given by

$$-\frac{1}{2}\left(\frac{\partial^2 W}{\partial A_i \partial A_j}\Psi_i\Psi_j + H.c.\right)$$

which in the LanHEP language will take form

```
lterm - df(W,H1,H2)*fH1*fH2 - ...   + AddHermConj.
```

where fH1, fH2 should be defined above as fermionic partners of corresponding multiples, e.g.

```
let f_h1 = { Zn31*up(~o1)+Zn32*up(~o2)+Zn33*up(~o3)+Zn34*up(~o4),
             Zm21*up('~1-')+Zm22*up('~2-') }.
```

Using the superpotential formalism in the MSSM and its extensions

- **FF* term from scalar supersymmetric potential**

$$V = \frac{1}{2} D^a D^a + F_i^\star F_i \quad \text{where} \quad F_i = \partial W / \partial A_i$$

in LanHEP notation will take a form

```
lterm - df(W,H1)*df(Wc,H1c) - ....
```

where Wc ***should be decleared above as the conjugate superpotential***

FF* term can be introduced even in shorter way as

```
lterm - dfdfc(W,H1) - ....
```

where dfdfc(W,H1) function evaluates the variational derivative, multiplies it by the conjugate expression and returns the result

CalcHEP: Calculator for High Energy Physics

born as a CompHEP in 1989: MSU-89-63/140

Comput.Phys.Commun. 184 (2013) 1729-1769, 1207.6082 [hep-ph]

Alexander Pukhov, AB, Neil Christensen http://theory.npi.msu.su/~pukhov/calchep.html

- **CalcHEP** is a powerful tool for efficient and effective studies of high-energy physics (HEP) phenomenology in Beyond the Standard Model (BSM) scenarios. It enables a high level of automation in going from your favourite model to physical observables such as decay widths, branching ratios, cross sections, kinematic distributions, and parton-level events.
- **Highlights**
  - Convenient graphical interface – to understand process in details
  - Output of symbolic results (Mathematica, REDUCE, FORM formats)
  - Calculates particle widths 'on the fly'
  - Easy to modify an existing model (GUI) or to implement the new one (LanHEP, FeynRules)
  - Batch interface: multidimensional scan of the the parameter space, produces LHE files in one run
  - Designed to study physics and present and future colliders: LHA PDF, ISR+Beamstrahlung for ILC
  - Modular structure: one can use it as a matrix element generator in other codes and packages (e.g.GAMBIT)
  - Has different modules for user modifications: user-defined cuts, user form factor etc.

CalcHEP installation guide

1. Download the code
a) **from Dark Tools github**
wget https://raw.githubusercontent.com/dimauromattia/darktools/main/calchep/calchep_3.9.2.tgz
or
b) **from the HEP Tools site**
→ Go to HEP TOOLS → calchep → download calchep_3.9.2.tgz
or
c) **from** http://theory.npi.msu.su/~pukhov/calchep.html
wget https://theory.sinp.msu.ru/~pukhov/CALCHEP/calchep_3.9.2.tgz

2. Unpack the archive: tar -zxvf calchep_3.9.2.tgz

3. Enter the directory: cd calchep_3.9.2

4. Compile: make

5. Start CalcHEP: cd work; ./calchep

ex#1 *install CalcHEP*

Compilation, potential problems and solutions

- To compile the CalcHEP source code you need:
  C compiler, the X11 graphics library and the X11 include files
  "CalcHEP is compiled successfully and can be started " is a good sign

- Supported operating sytems: Linux, MacOS, SunOS, and Windows Subsystem for Linux (WSL)

- Potential problem
  - The most frequent compilation problem is due to the absence of the X11 include files: in this case CalcHEP will compile, however, it only runs in non-interactive mode: ./calchep will give
    Error: You have launched the interactive session for a version
    of CalcHEP that has been compiled without the X11 library.
    Presumably, the X11 development package is not installed on your computer.
  - the following additional package should be installed to run CalcHEP in GUI mode:
    libX11-devel on Fedora/Scientific; libX11-dev on Ubuntu/Debian; xorg-x11-devel on SUSE;
    for MAC: XQuartz is the official X11 server and client-side libraries for macOS
    get it from https://www.xquartz.org or brew install --cask xquartz
- Compilation for High Precision Calculations
  - Intel C compiler has a _Quad type, -D QUAD has to be added to FlagsForSh as
    CFLAGS="-D_QUAD_ -fPIC -fsigned-char -Qoption,cpp,--extended_float_type"

Features/Limitations of CalcHEP

- Can evaluate any decay and scattering processes within any (user defined) model!
- Tree-level processes
- Squared Matrix Element calculation
  no spin information for outgoing particles – spin averaged amplitude
- Limit on number of external legs (involved particles) and number of diagrams
  - official limit – 8 , unofficial – none
  - limit is set from the practical point of view:
    - 2 → 6 (1→7) set the essential time/memory limit
    - number of diagrams ~ 500 set the disk space and the time limit

http://theory.npi.msu.su/~pukhov/calchep.html

CalcHEP - a package for calculation of Feynman diagrams and integration over multi-particle phase space.

Authors - Alexander Pukhov, Alexander Belyaev, Neil Christensen

The main idea of CalcHEP is to enable one to go directly from the Lagrangian to the cross sections and distributions effectively, with a high level of automation. The package can be compiled on any Unix platform.

General information
Main features , Acknowledgments Publications&Lectures Contributions

Manual
calchep_man_3.3.6.pdf (manual for version 3.3.6, July 19, 2012)
HEP computer tools (Lecture by Alexander Belyaev)
See also: Dan Green, High Pt physics at hadron colliders (Cambrige University Press)

manual

Code download.
License GPL-3
Installation Current version 3.9.2(updated 05.03.2025) New Options and Bugs Fixed All versions

new options and writeup!
arXiv:1207.6082

Models:
MSSM_10.14(15.10.2014) NMSSM_8.15(25.08.2015) CPVMSSM_10.14(16.10.2014) SUSY models By A.Semenov Lepto
Model database HEPMDB

Related packages on Web:
Packages for model generation: LanHEP FeynRules SARAH
RGE and spectrum calculation: SuSpect Isajet SoftSUSY SPheno CPsuperH NMSSMTools
Particle widths in MSSM: SUSY-HIT HDECAY
Parton showers: PYTHIA

Contacts

Email: **calchep@googlegroups.com** *Launchpad service:* *Ask a question* *File a bug*

Connected to launchpad system

./calchep

CalcHEP_3.9.2/symb

```
   CalcHEP - a package for Calculation in High Energy Physics
           Version 3.9.2: Last correction  March 5, 2025


 Authors: Alexander Pukhov(Skobeltsyn Institute of Nuclear Physics,Moscow)
          Alexander Belyaev(University of Southampton)
          Neil Chistensen (University of Pittsburgh)

 For contacts:   email       : <calchep@googlegroups.com>
                 Questions   : https://answers.launchpad.net/calchep
                 Bugs        : https://bugs.launchpad.net/calchep
                 Code&Models: http://theory.sinp.msu.ru/~pukhov/calchep.html
 The BSMs for CalcHEP were developed in collaboration with:
       G.Belanger,F.Boudjema,A.Semenov

 The package contains codes written by:
       M.Donckt,V.Edneral,V.Ilyin,D.Kovalenko,A.Kryukov,G.Lepage,A.Semenov

                  Press F9 or click the box below to get

                 References, Contributions, Acknowledgments

This information is available during the session by means of the F9 key
```

CalcHEP structure/modes

- Graphical mode
  - symbolic part
  - numerical part
- Batch mode

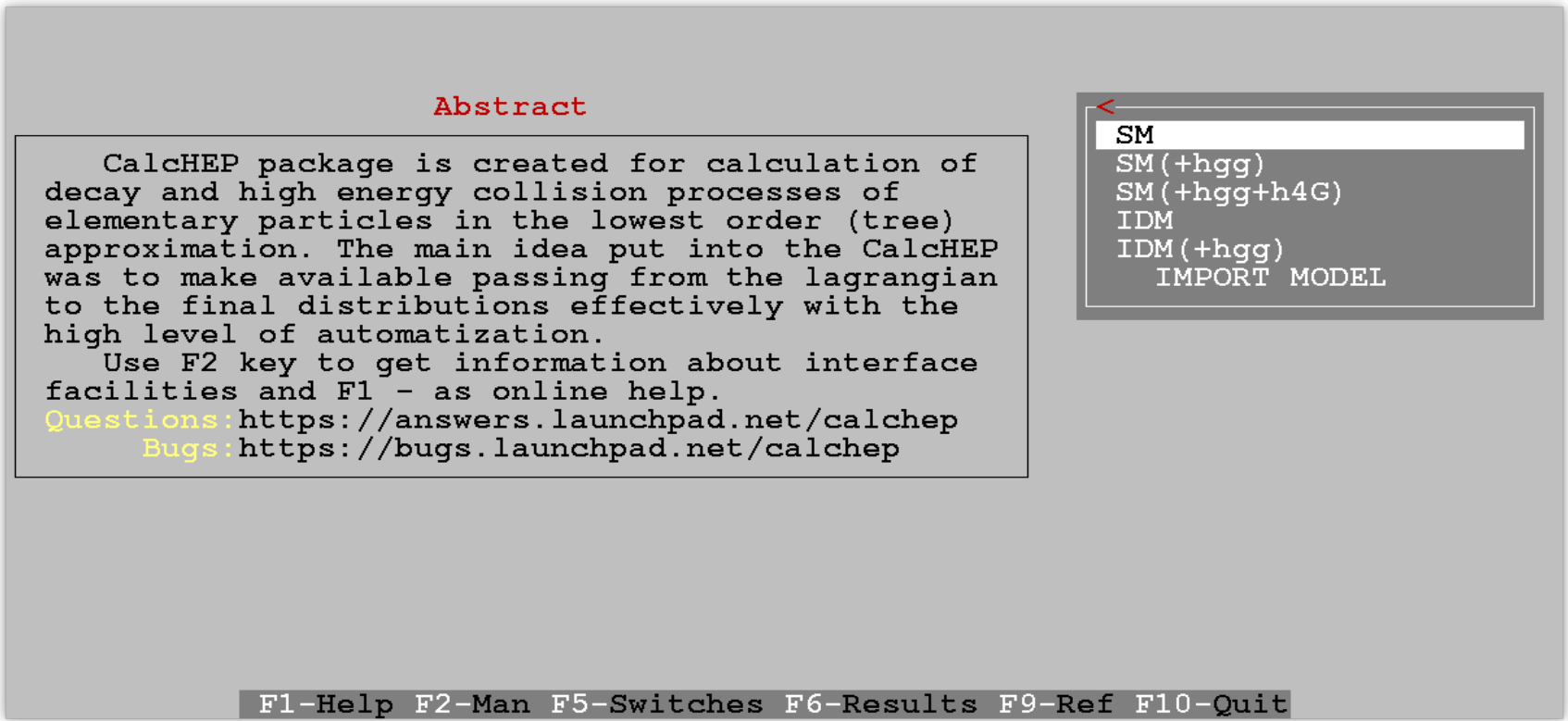
Principle KEYS for CalcHEPs GUI

Enter menu selection (forward)

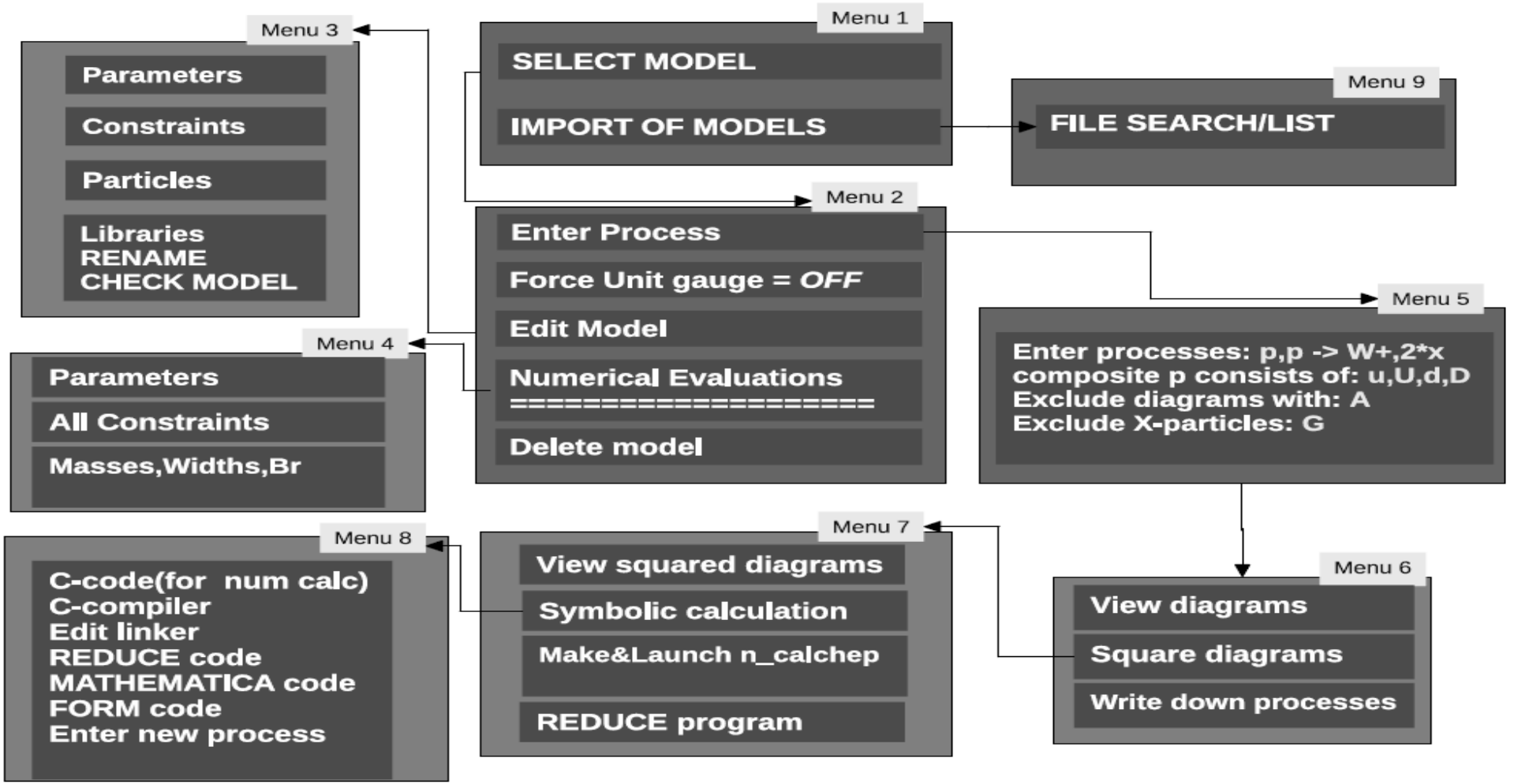
Exit menu selection (back)

Help!

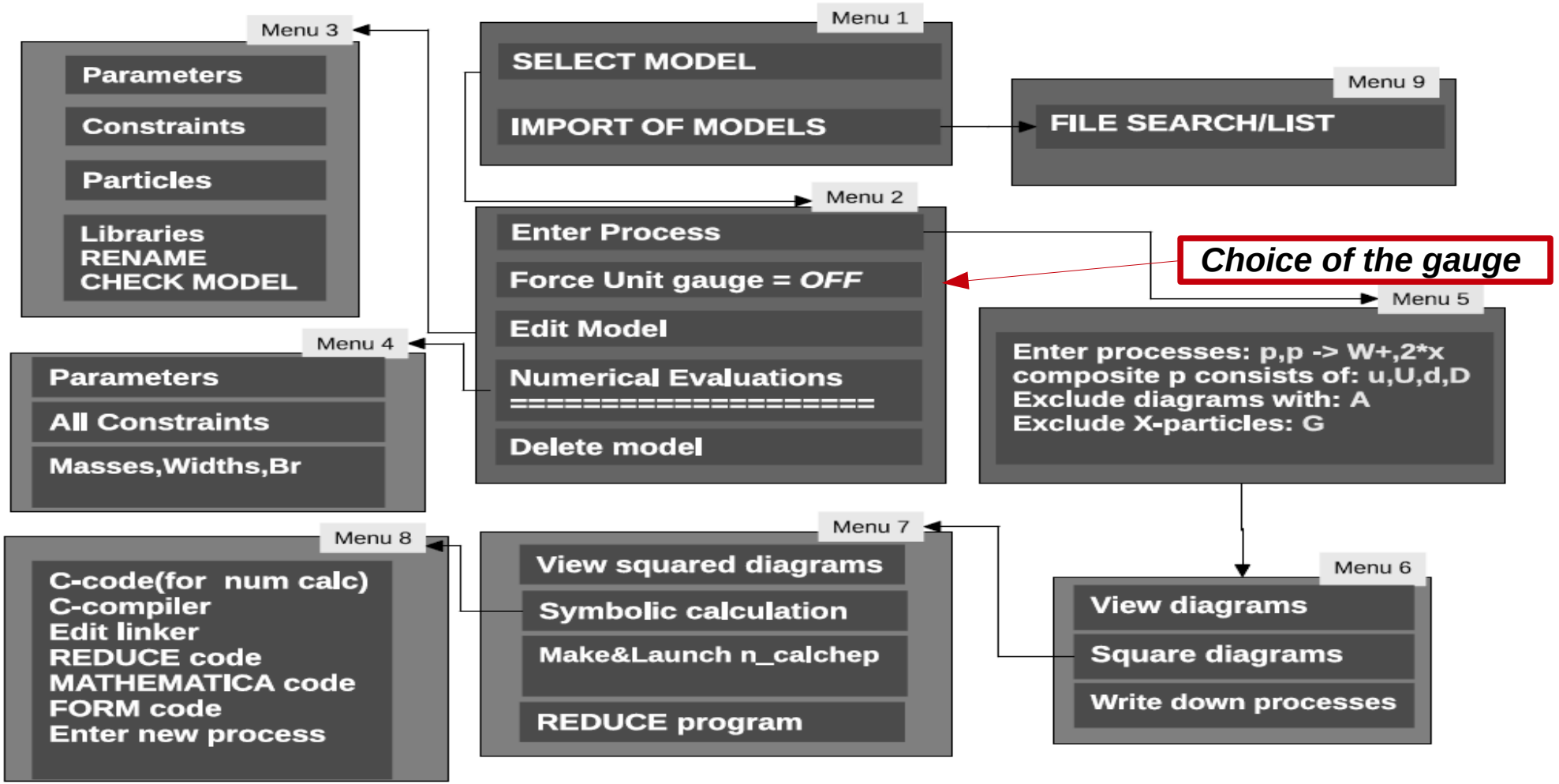
Starting CalcHEP



CalcHEP menu structure: symbolic part

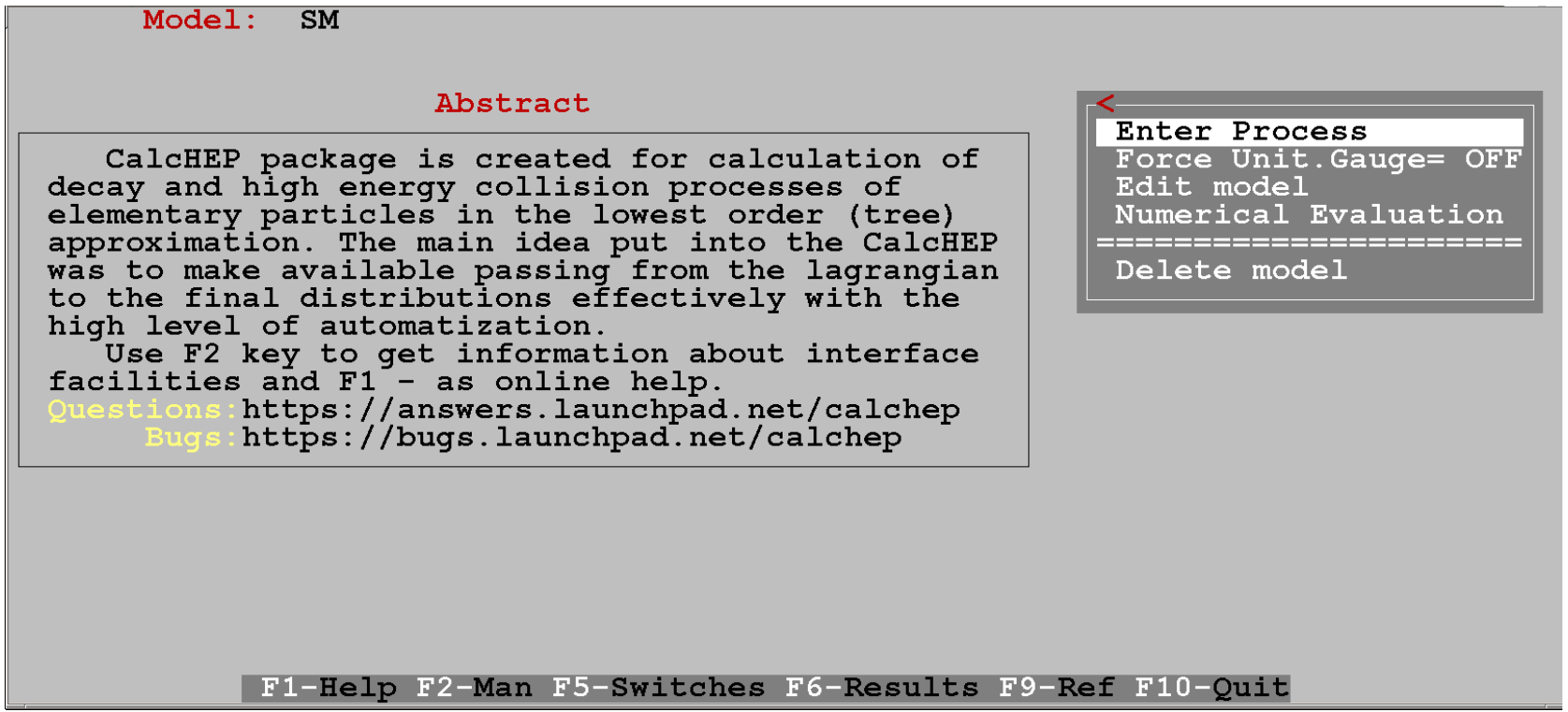


CalcHEP menu structure: symbolic part



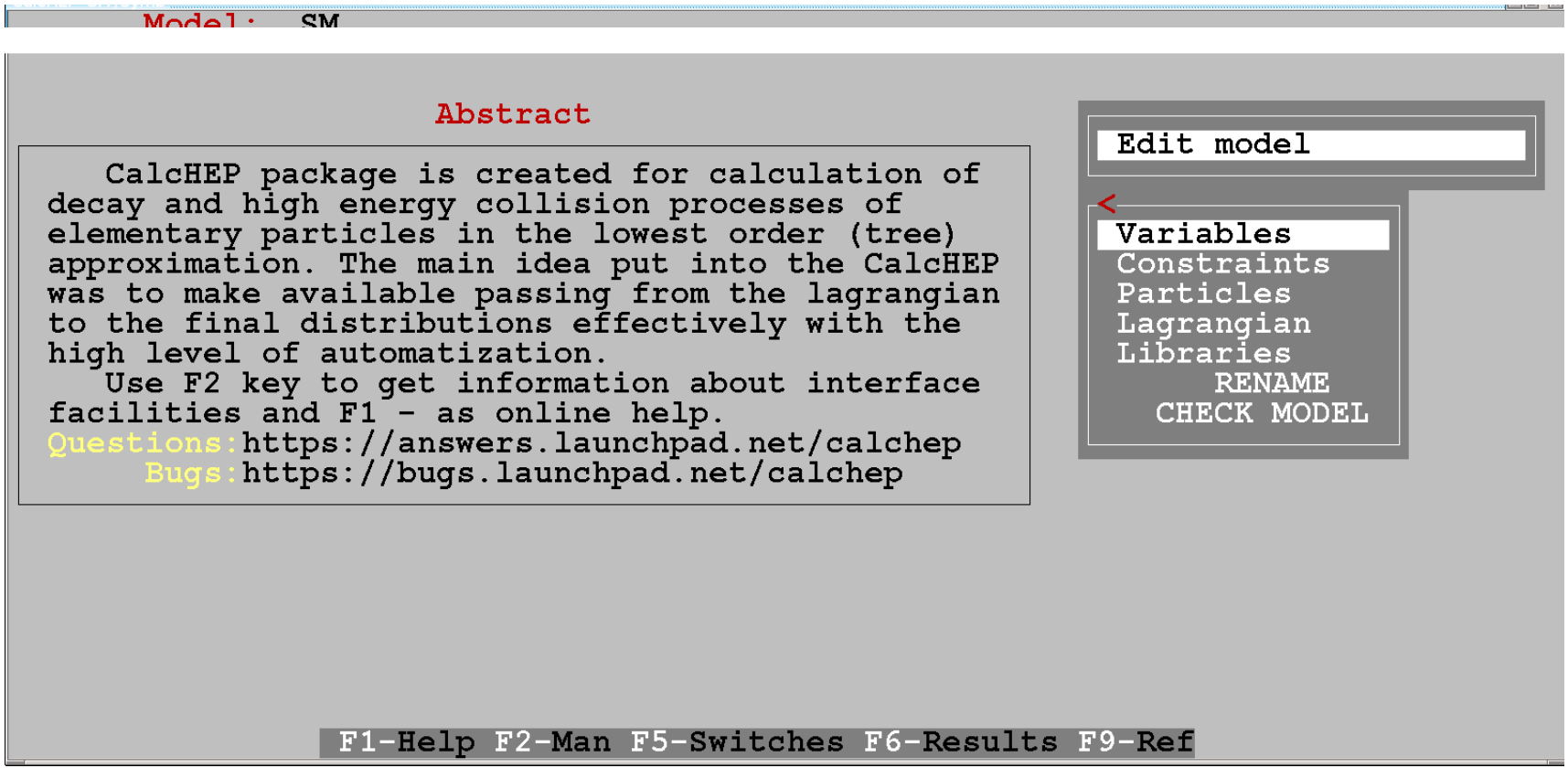
Model choice and Process input

Choose your gauge ***Enter Process***
Edit Model ***Numerical Evaluation***



The Model Structure

Parameters ***Constraints***
Particles ***Vertices***



Particles: prtcl*xx*.mdl (spins 0,1/2,1,3/2,2)

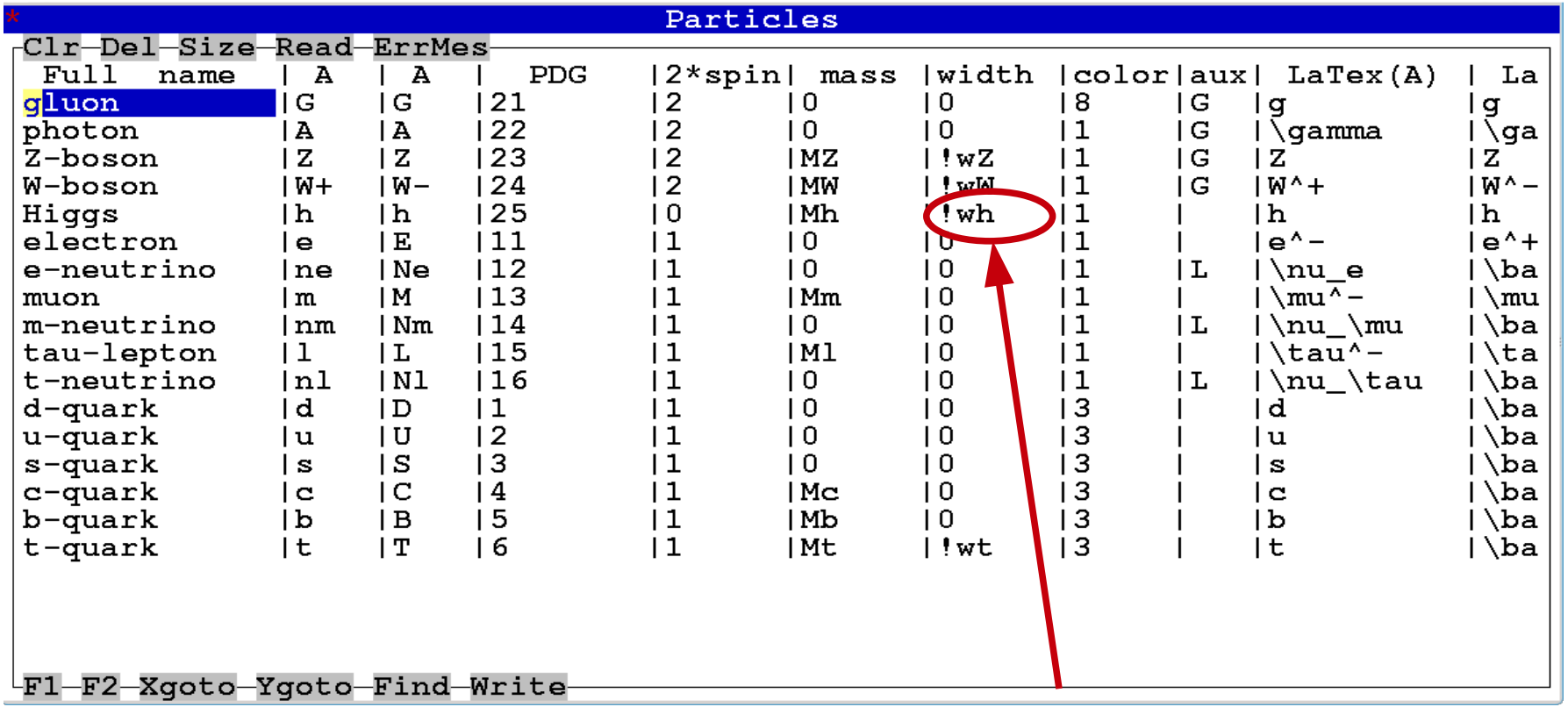
Particles

Clr Del Size Read ErrMes

Full name	A	A	PDG	2*spin	mass	width	color	aux	LaTex(A)	La
gluon	G	G	21	2	0	0	8	G	g	g
photon	A	A	22	2	0	0	1	G	\gamma	\ga
Z-boson	Z	Z	23	2	MZ	!wZ	1	G	Z	Z
W-boson	W+	W-	24	2	MW	!wW	1	G	W^+	W^-
Higgs	h	h	25	0	Mh	!wh	1		h	h
electron	e	E	11	1	0	0	1		e^-	e^+
e-neutrino	ne	Ne	12	1	0	0	1	L	\nu_e	\ba
muon	m	M	13	1	Mm	0	1		\mu^-	\mu
m-neutrino	nm	Nm	14	1	0	0	1	L	\nu_\mu	\ba
tau-lepton	l	L	15	1	Ml	0	1		\tau^-	\ta
t-neutrino	nl	Nl	16	1	0	0	1	L	\nu_\tau	\ba
d-quark	d	D	1	1	0	0	3		d	\ba
u-quark	u	U	2	1	0	0	3		u	\ba
s-quark	s	S	3	1	0	0	3		s	\ba
c-quark	c	C	4	1	Mc	0	3		c	\ba
b-quark	b	B	5	1	Mb	0	3		b	\ba
t-quark	t	T	6	1	Mt	!wt	3		t	\ba

F1 F2 Xgoto Ygoto Find Write

Particles: prtcl*xx*.mdl



Particles

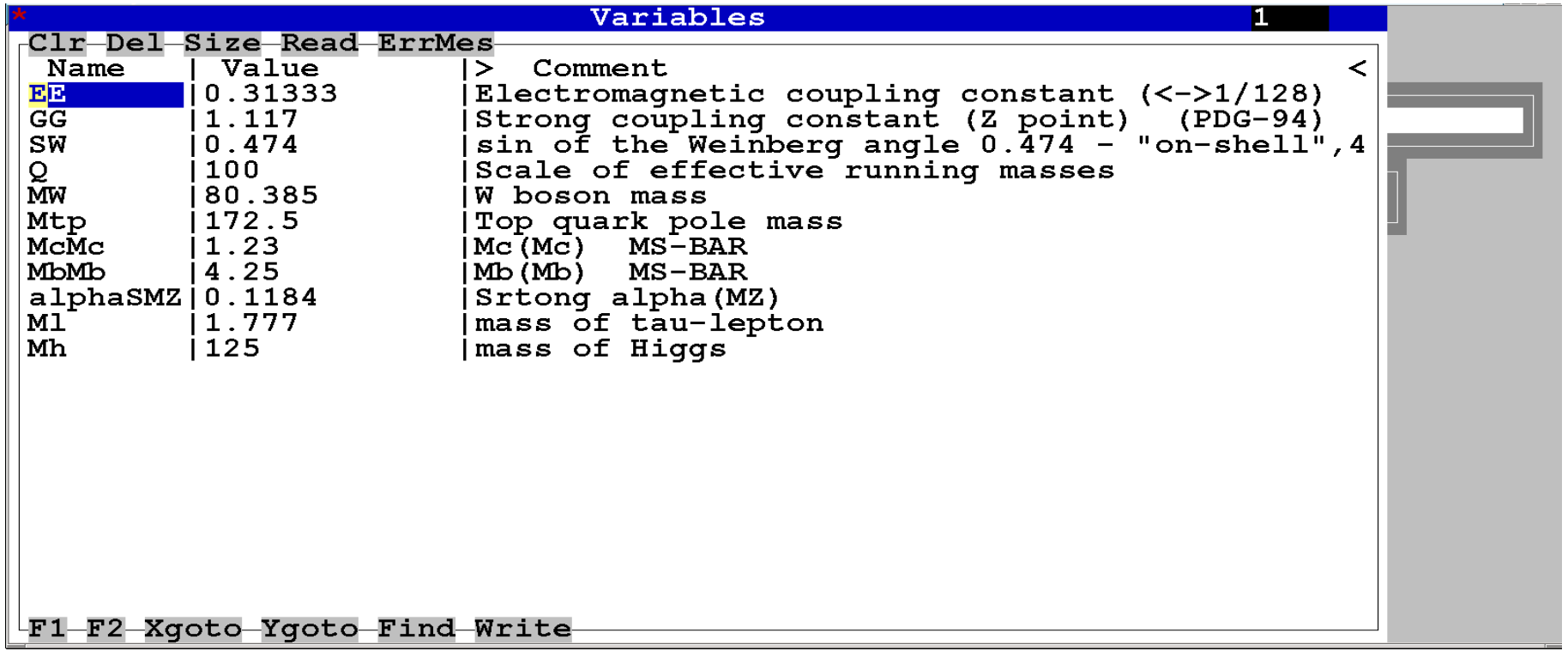
Clr Del Size Read ErrMes

Full name	A	A	PDG	2*spin	mass	width	color	aux	LaTex(A)	La
gluon	G	G	21	2	0	0	8	G	g	g
photon	A	A	22	2	0	0	1	G	\gamma	\ga
Z-boson	Z	Z	23	2	MZ	!wZ	1	G	Z	Z
W-boson	W+	W-	24	2	MW	!wW	1	G	W^+	W^-
Higgs	h	h	25	0	Mh	!wh	1		h	h
electron	e	E	11	1	0	0	1		e^-	e^+
e-neutrino	ne	Ne	12	1	0	0	1	L	\nu_e	\ba
muon	m	M	13	1	Mm	0	1		\mu^-	\mu
m-neutrino	nm	Nm	14	1	0	0	1	L	\nu_\mu	\ba
tau-lepton	l	L	15	1	Ml	0	1		\tau^-	\ta
t-neutrino	nl	Nl	16	1	0	0	1	L	\nu_\tau	\ba
d-quark	d	D	1	1	0	0	3		d	\ba
u-quark	u	U	2	1	0	0	3		u	\ba
s-quark	s	S	3	1	0	0	3		s	\ba
c-quark	c	C	4	1	Mc	0	3		c	\ba
b-quark	b	B	5	1	Mb	0	3		b	\ba
t-quark	t	T	6	1	Mt	!wt	3		t	\ba

F1 F2 Xgoto Ygoto Find Write

Higgs boson width will be calculated `on the fly`

Independent parameters: varsxx.mdl



Dependent parameters(constraints): func*xx*.mdl

Constraints

Name	> Expression
CW	sqrt(1-SW^2) % cos of the Weinberg angle
GF	EE^2/(2*SW*MW)^2/Sqrt2 % experimental value 1.166E-5 [1/GeV^2]
MZ	MW/CW % Z boson mass
LamQCD	initQCD5(alphaSMZ, McMc, MbMb, Mtp)
Mb	MbEff(Q)
Mc	McEff(Q)
Ms	MqEff(0.096, Q) % s-quark effective mass via 2MeV running one
LAAh	-cabs(lAAhiggs(Mh, "h"))
LGGh	-cabs(lGGhiggs(Mh, "h"))
aQCDh	alphaQCD(Mh)/acos(-1)
RQCDh	sqrt(1+149/12*aQCDh+68.6482*aQCDh^2-212.447*aQCDh^3)
B00000	1-2*SW^2
B00001	1-4*SW^2+4*SW^4

Clr Del Size Read ErrMes

F1 F2 Xgoto Ygoto Find Write

Feynman rules: lgrng*xx*.mdl

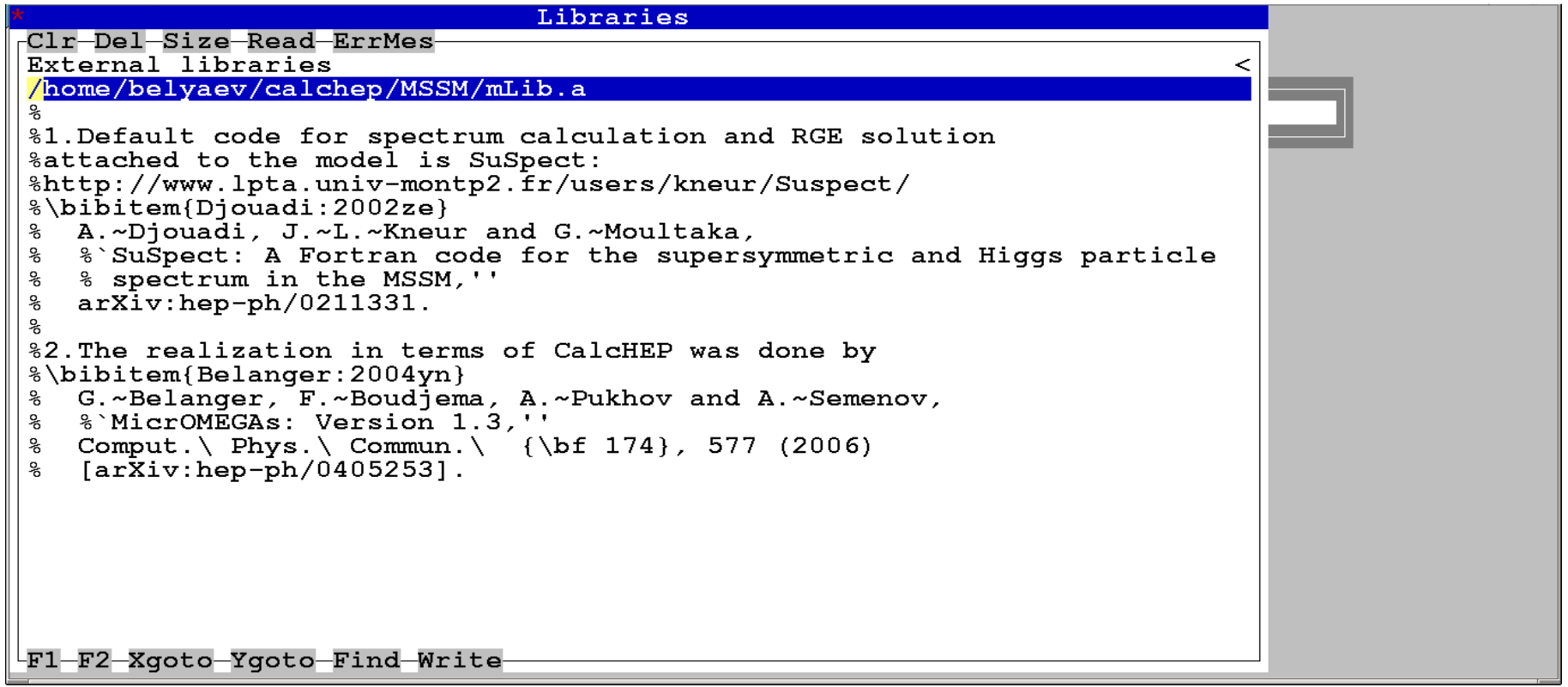
Lagrangian 1

Clr	Del	Size	Read	ErrMes	P1	P2	P3	P4	> Factor <	> dLagrangian/ dA(p1) dA(p2) dA(p3)
					A	A	h		-4*LAAh	p1.p2*m1.m2-m2.p1*m1.p2
					A	W+	W-		EE	m3.p2*m1.m2-m1.p2*m2.m3-m2.p3*m1.m3+m
					A	W+	W-.f		-i*EE*MW	m1.m2
					A	W+.f	W-		i*EE*MW	m1.m3
					A	W+.f	W-.f		-EE	m1.p3-m1.p2
					A.C	W+.c	W-		EE	m3.p1
					A.C	W-.c	W+		-EE	m3.p1
					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	Z.f		i*EE*Mb/(2*MW*SW)	G5
					B	b	h		-EE*Mb/(2*MW*SW)	1
					B	t	W-		EE*Sqrt2/(4*SW)	G(m3)*(1-G5)
					B	t	W-.f		i*EE*Sqrt2/(4*MW*SW)	Mb*(1-G5)-Mtp*(1+G5)
					C	c	A		2*EE/3	G(m3)
					C	c	G		GG	G(m3)
					C	c	Z		-EE/(12*CW*SW)	8*SW^2*G(m3)-3*G(m3)*(1-G5)
					C	c	Z.f		-i*EE*Mc/(2*MW*SW)	G5
					C	c	h		-EE*Mc/(2*MW*SW)	1
					C	s	W+		EE*Sqrt2/(4*SW)	G(m3)*(1-G5)
					C	s	W+.f		-i*EE*Sqrt2/(4*MW*SW)	Ms*(1+G5)-Mc*(1-G5)
					D	d	A		-EE/3	G(m3)
					D	d	G		GG	G(m3)
					D	d	Z		EE/(12*CW*SW)	4*SW^2*G(m3)-3*G(m3)*(1-G5)

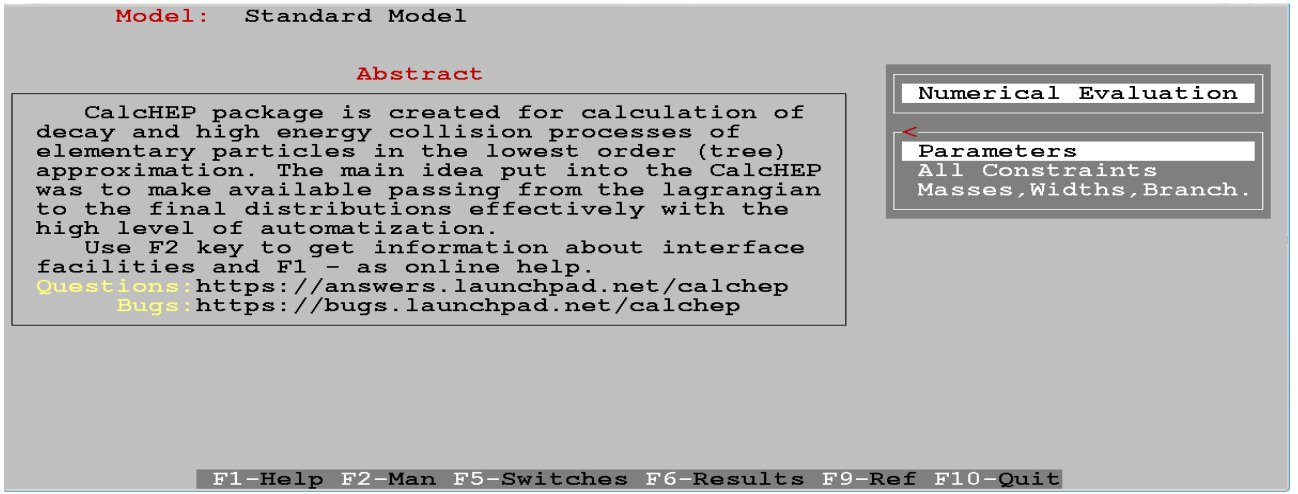
F1 F2 Xgoto Ygoto Find Write

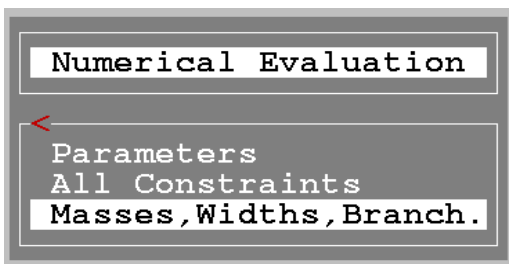
External Libraries: extlib*xx*.mdl

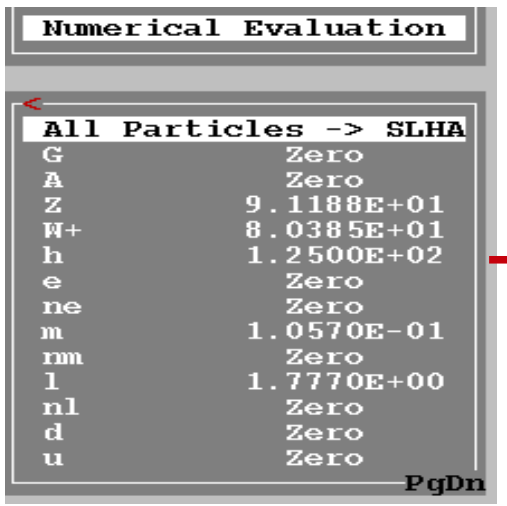
Typically is empty for simple models but can be used for any library which helps to build complicated model. E.g. mass spectra calculator for SUSY (involving RGE solutions etc)



Numerical evaluation of masses & branchings







ex#3: Find the SM particles spectrum and Br ratios

Syntax for the process

- **the input syntax: P1[,P2] -> P3,P4 [,,...,[N*x]]**

- **hadron/composite particle scattering**

  'p*,p*->W+,b,B'

  unknown particle are assumed to be composite,

  if you use 'p*', the u,U,d,D,s,S,c,C,b,B,G structure will be used automatically

- **wild cards/names for outgoing particles**

  'H -> 2*x'

- **intermediate particles can be non-trivially excluded**

  'W+ > 2, A>1, Z>3'

ex#4: SM Higgs production cross section for e+e- → HZ process versus the collider energy for 0.5-1.0 TeV range, 10 GeV step

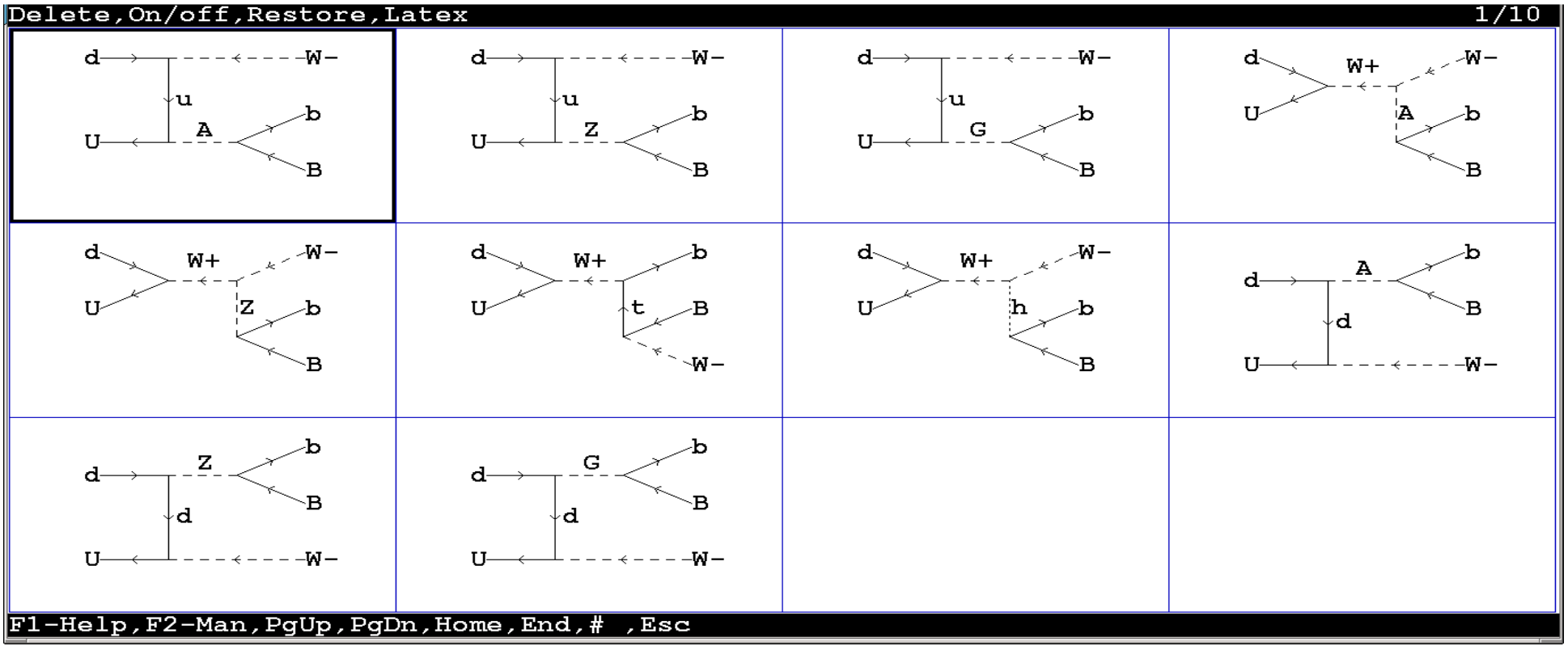
Symbolic session(1)

```
      Model:   SM

            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


Enter  process: p*,p*->W,b,B
composite 'p*'  consists of: G,d,D,u,U,s,S,c,C,b,B
composite 'W'  consists of: W+,W-
Exclude diagrams with
```

Symbolic session (2)

```
        Model:   SM

      Process:   p*,p*->W,b,B

                 Feynman diagrams                    View diagrams
88      diagrams in  8    subprocesses  are constructed.
0       diagrams  are deleted.

  NN          Subprocess                                  Del    Rest
  <
     1| d,U -> W-,b,B                                     |    0|    10
     2| D,u -> W+,b,B                                     |    0|    10
     3| u,D -> W+,b,B                                     |    0|    10
     4| U,d -> W-,b,B                                     |    0|    10
     5| s,C -> W-,b,B                                     |    0|    12
     6| S,c -> W+,b,B                                     |    0|    12
     7| c,S -> W+,b,B                                     |    0|    12
     8| C,s -> W-,b,B                                     |    0|    12

 F1-Help F2-Man F3-Model F5-Switches F6-Results F7-Del F8-UnDel F9-Ref F10-Quit
```

Symbolic session (3)



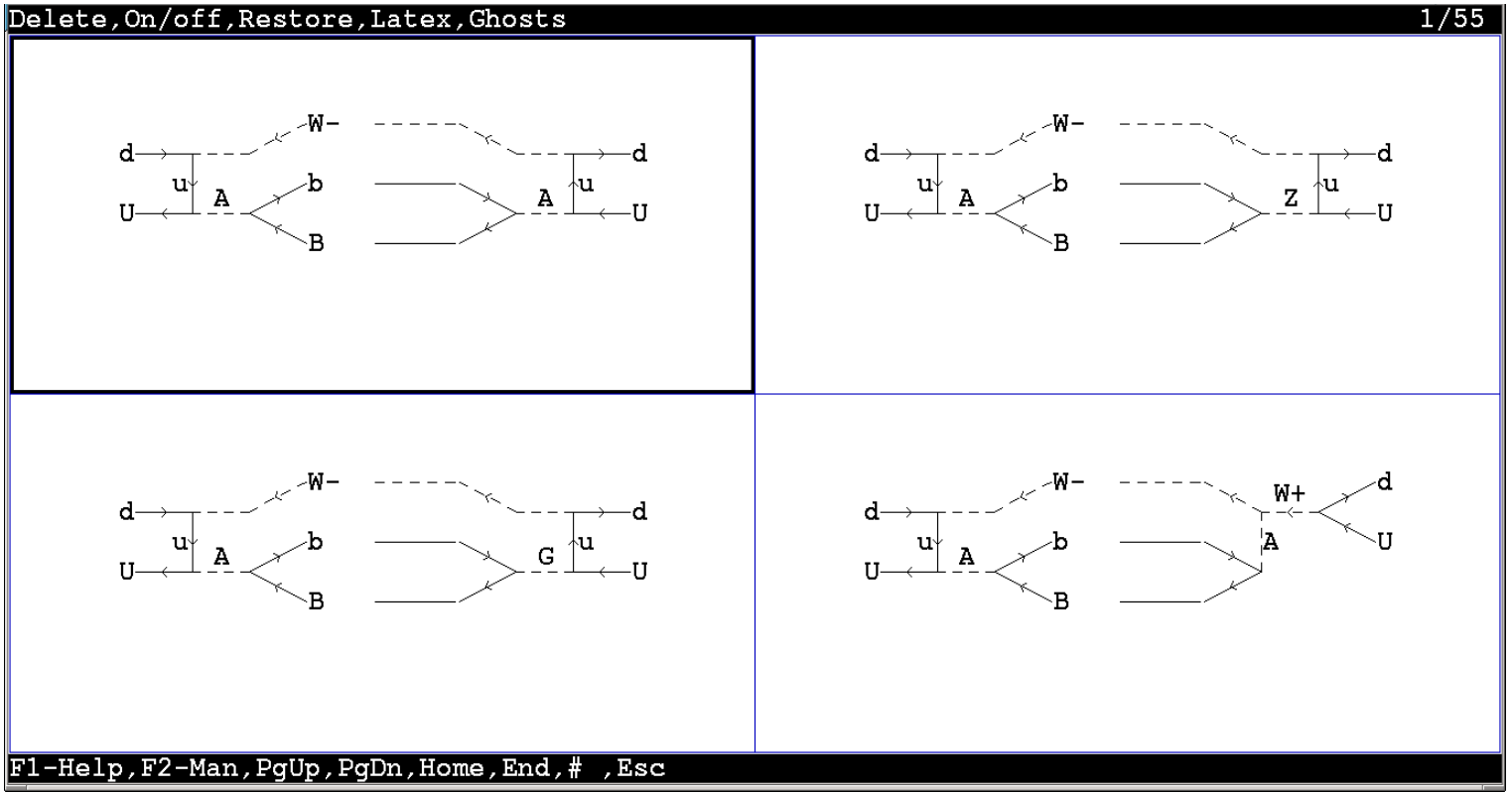
Symbolic session (4)

```
       Model:   SM

     Process:   p*,p*->W,b,B

                Feynman diagrams                    View squared diagrams
88     diagrams in   8    subprocesses   are constructed.
0      diagrams   are deleted.

                Squared diagrams
532    diagrams in   8    subprocesses   are constructed.
0      diagrams   are deleted.
0      diagrams   are calculated.
```

NN	Subprocess	Del	Calc	Rest
1	d,U->W-,b,B	0	0	55
2	D,u->W+,b,B	0	0	55
3	u,D->W+,b,B	0	0	55
4	U,d->W-,b,B	0	0	55
5	s,C->W-,b,B	0	0	78
6	S,c->W+,b,B	0	0	78
7	c,S->W+,b,B	0	0	78
8	C,s->W-,b,B	0	0	78

```
F1-Help F2-Man F3-Model F4-Diagrams F5-Switches F6-Results F9-Ref F10-Quit
```

Symbolic session (5)



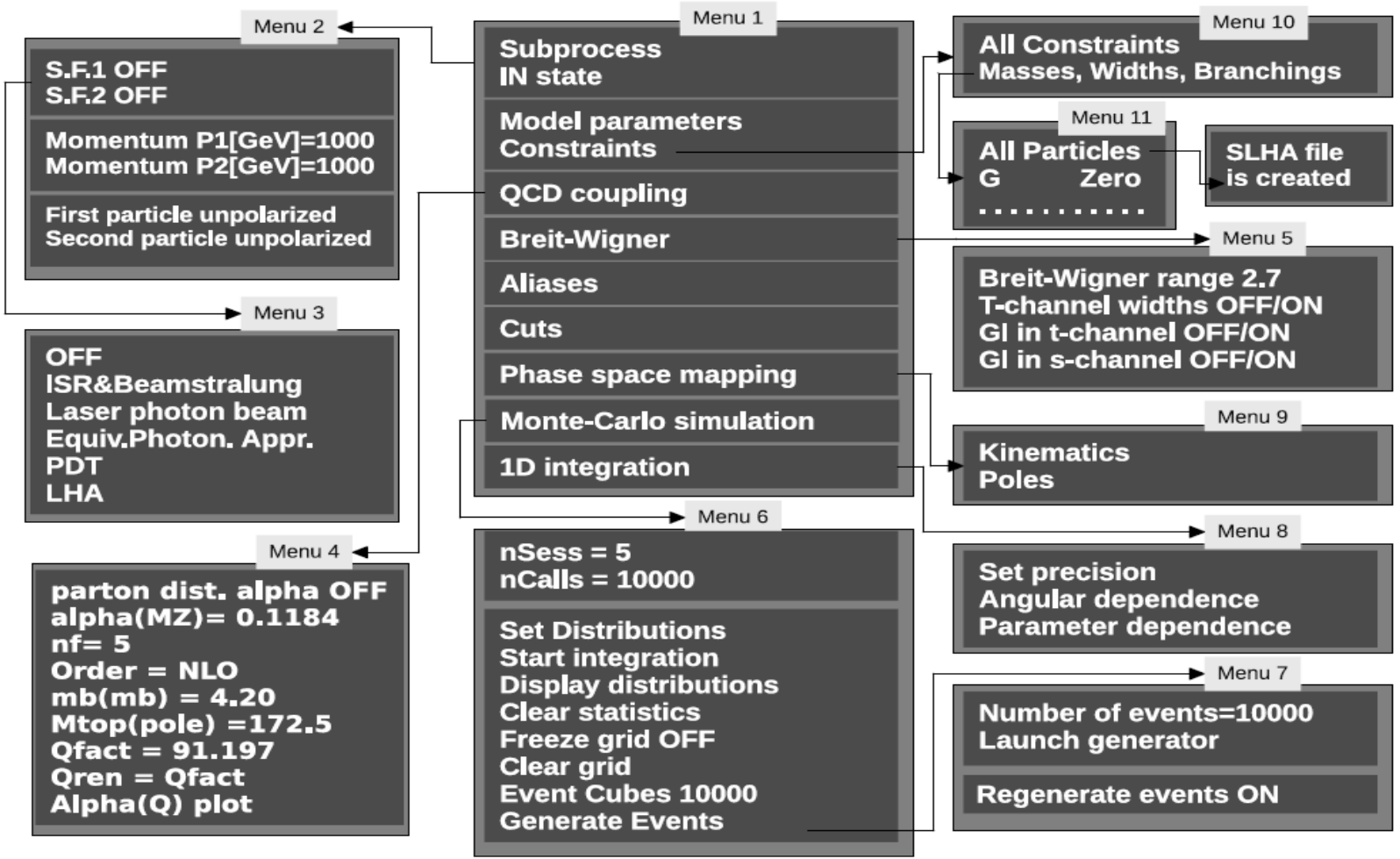
Symbolic session (6)

```
      Model:   SM

    Process:   p*,p*->W,b,B

               Feynman diagrams
88     diagrams in  8   subprocesses  are constructed.
0      diagrams  are deleted.

               Squared diagrams
532    diagrams in  8   subprocesses  are constructed.
0      diagrams  are deleted.
532    diagrams  are calculated.
```

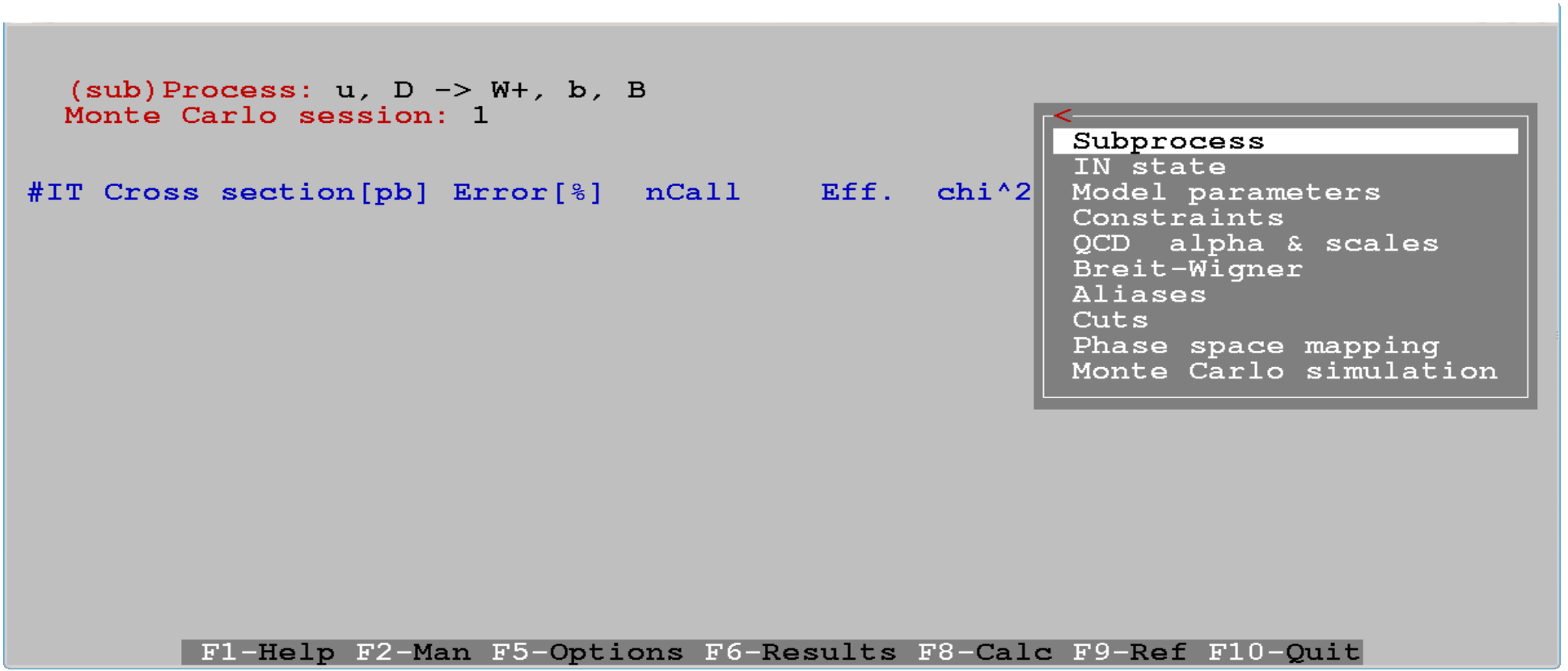
```
<
 C code
      C-compiler
      Edit Linker
 REDUCE code
 MATHEMATICA code
 FORM code
 Enter new process
```

```
 F1-Help F2-Man F3-Model F4-Diagrams F5-Switches F6-Results F9-Ref F10-Quit
```

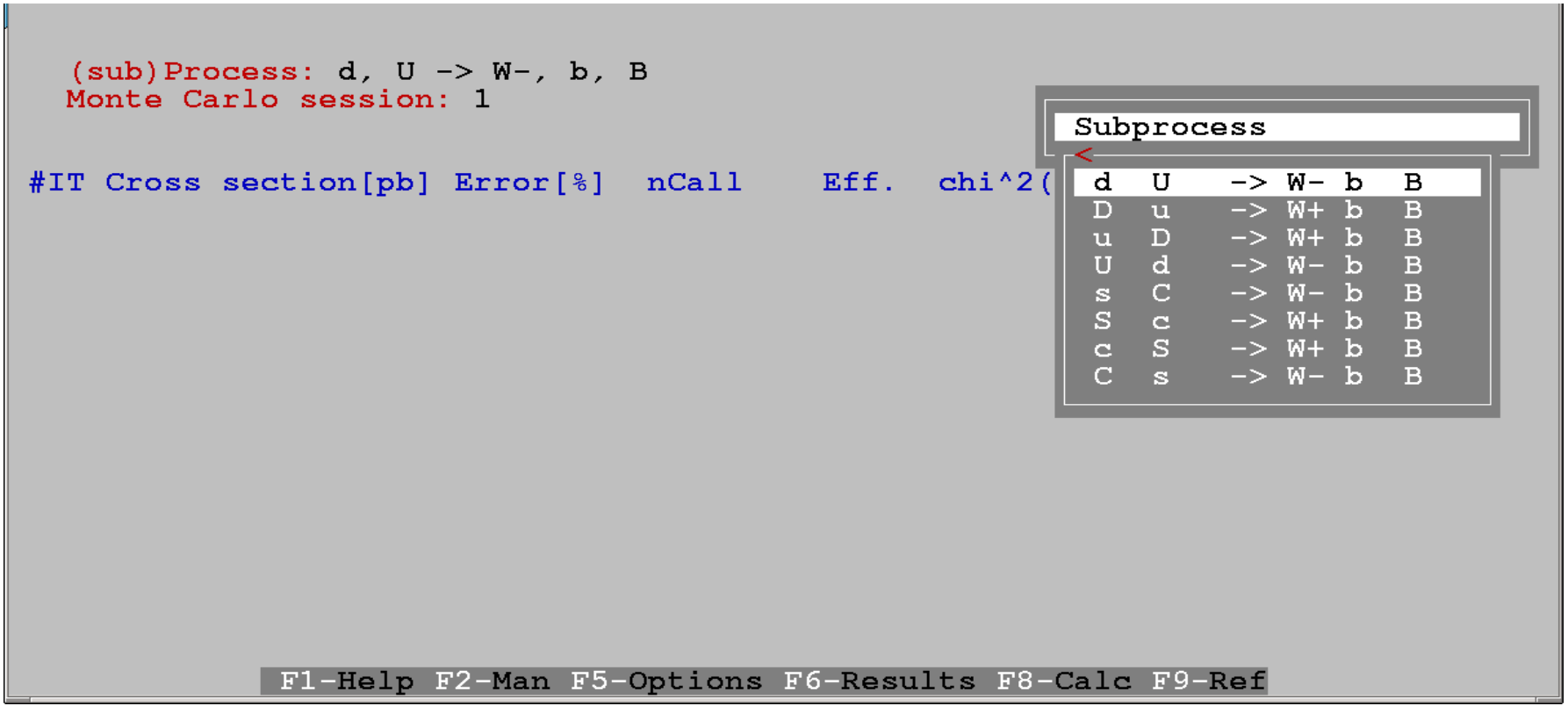
Menu structure of the numerical part



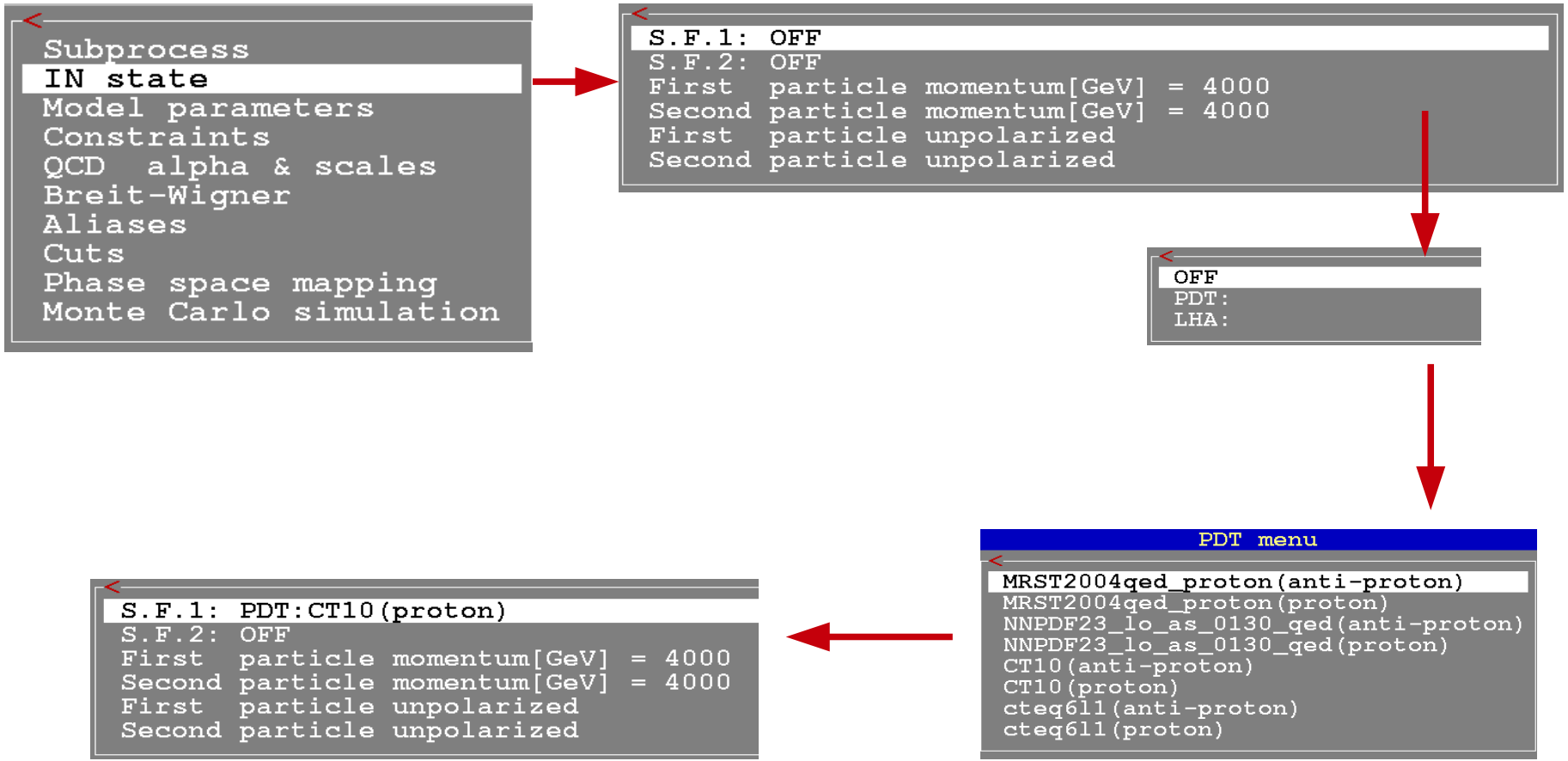
Numerical part(1)



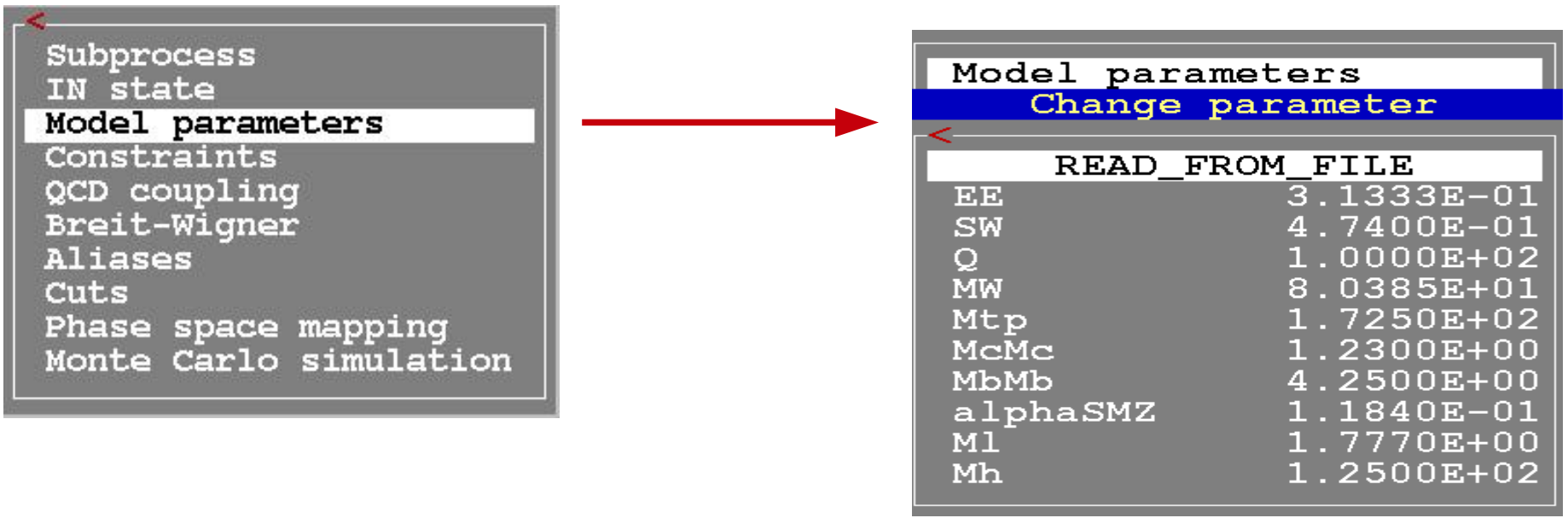
Numerical part(2)



control of the initial states and parton density functions



model parameters



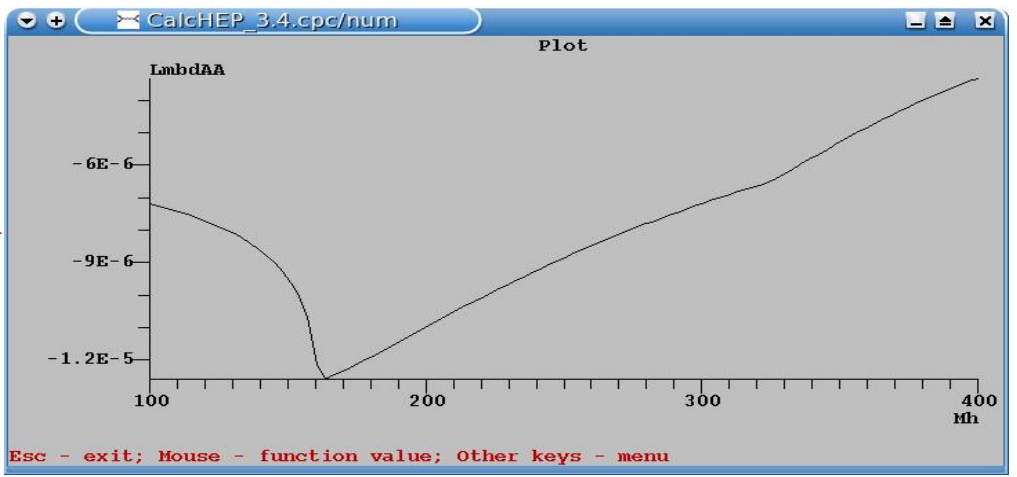
dependent parameters (SM+ggH model)

```
Subprocess
IN state
Model parameters
Constraints
QCD  alpha & scales
Breit-Wigner
Aliases
Cuts
Phase space mapping
Monte Carlo simulation
```

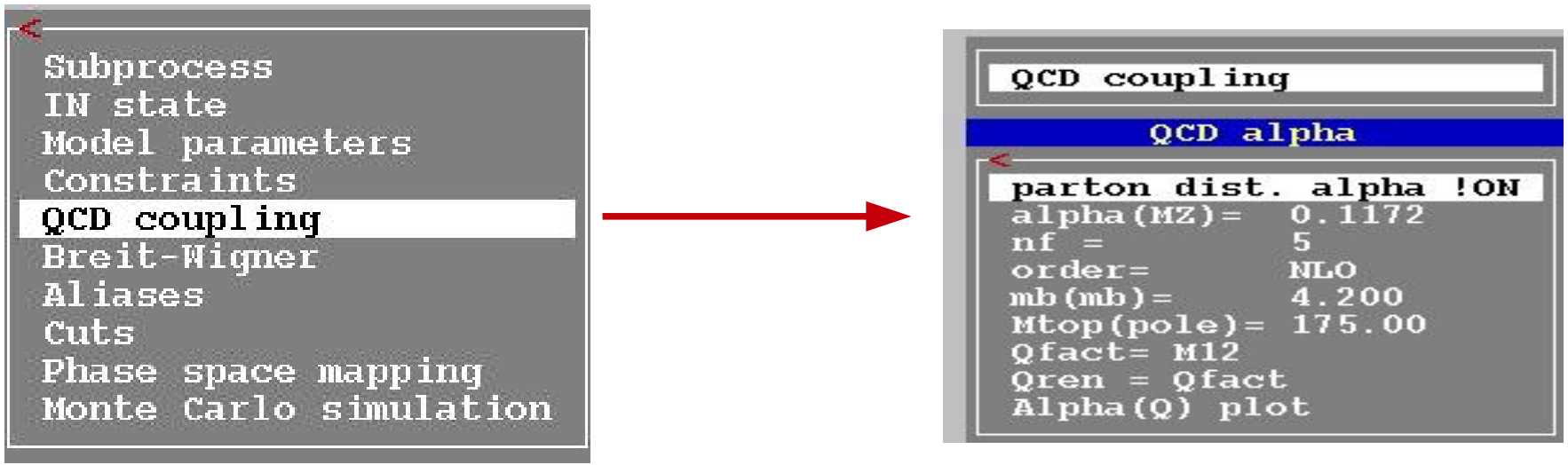
```
Constraints
All Constraints
Masses,Widths,Branching
```

```
Constraints
Display dependence
                    Print to file
CW                  8.8052E-01
GF                  1.1954E-05
MZ                  9.1292E+01
LamQCD              3.4641E-01
Mb                  3.1986E+00
Mc                  6.1137E-01
Ms                  5.9444E-02
LAAh               -7.8864E-06
LGGh               -1.3054E-05
aQCDh               3.5959E-02
```

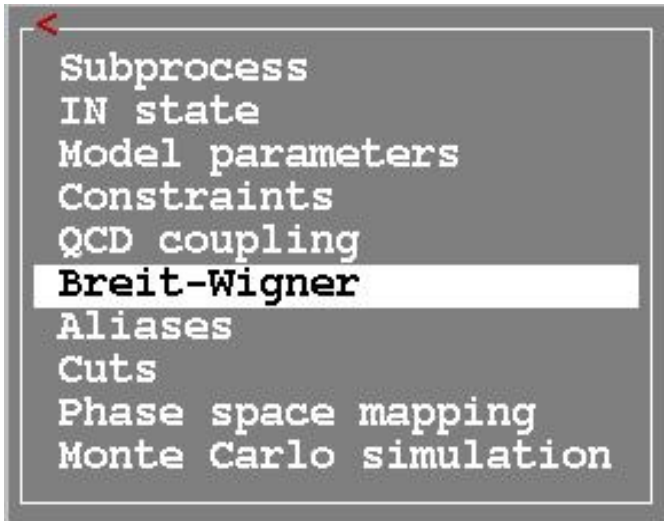
```
Constraints
Display dependence
LmbdAA          -7.8845E-06
on parameter
Mh               1.2500E+02
Plot
x-Min = 100
x-Max = 400
Npoints = 100
Display
```



QCD coupling and the QCD scale



control of resonances



F1

n_width_1

This menu sets value R which defines range of implementation of Breit-Wigner formula. Namely it is used in the region where

$|p^2-m^2| < R*m*w$

For region

$|p^2-m^2| > \sqrt{(R^2+1)}*m*w$

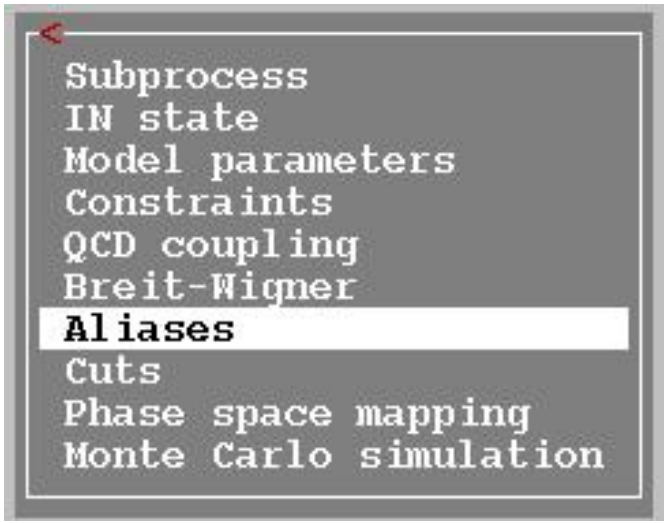
we use zero width propagator. In the intermediate region constant propagator interpolates both formulas.

In general Breit-Wigner leads to breaking of gauge invariance. In its turn it can lead to the lost of diagram cancellation. From the other side just in the point $p^2=m^2$ the contribution of pole

diagram have to be gauge invariant. Thus at this

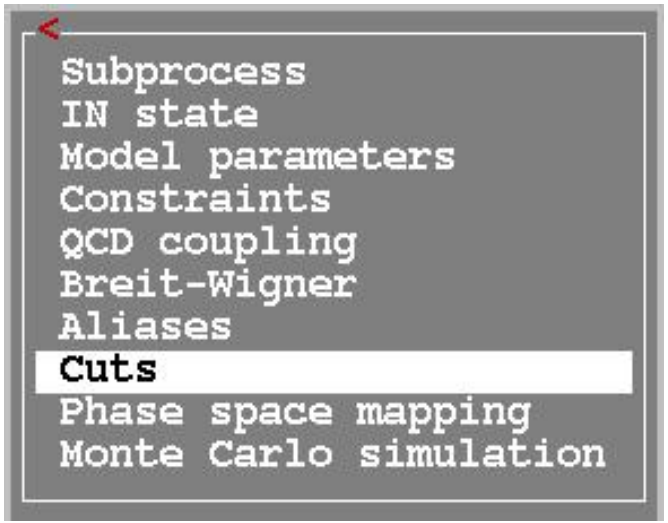
pont cancellation between pole and non-pole diagra ms
is not expected. We assume that close to pole the problem also is not so serious. But far from the p ole
we ignore width and restore gauge invariance.

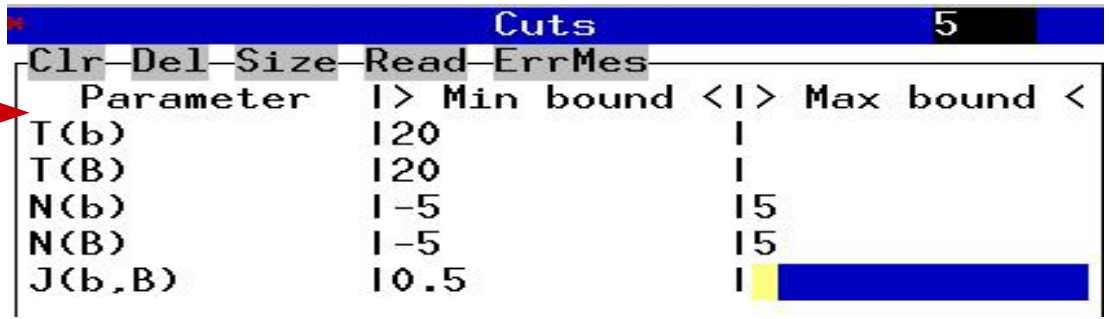
Aliases


Subprocess
IN state
Model parameters
Constraints
QCD coupling
Breit-Wigner
Aliases
Cuts
Phase space mapping
Monte Carlo simulation

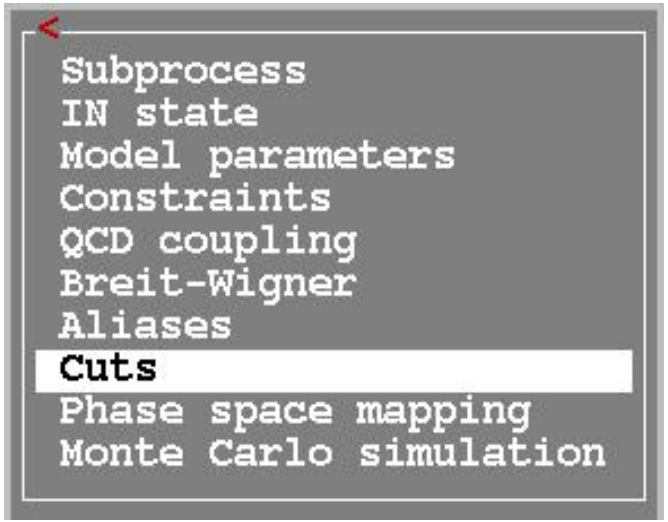

Composites
Clr Del Size Read ErrMes
Name |> Comma separated list of particles
Jet |u,U,d,D,s,S,c,C,G

setting kinematical cuts





setting kinematical cuts



Cuts 0

Clr Del Size Read ErrMes

Parameter |> Min bound <|> Max bound <

F1

n_cut

This table apples cuts on the phase space. A phase space function is described in the first column. Its limits are defined and the second and the third columns. If one of these fields is empty then a one-side cut is applied.

The phase space function is defined by its name which characterize type of cut and a particle list for which the cut is applied. For example, "T(u)" means transverse momentum of 'u'-quark; T(u,D) means summary transverse momentum of quark pair. The following cut functions are available:

A - Angle in degree units;
C - Cosine of angle;
J - Jet cone angle;
E - Energy of the particle set;
M - Mass of the particle set;
P - Cosine in the rest frame of pair;

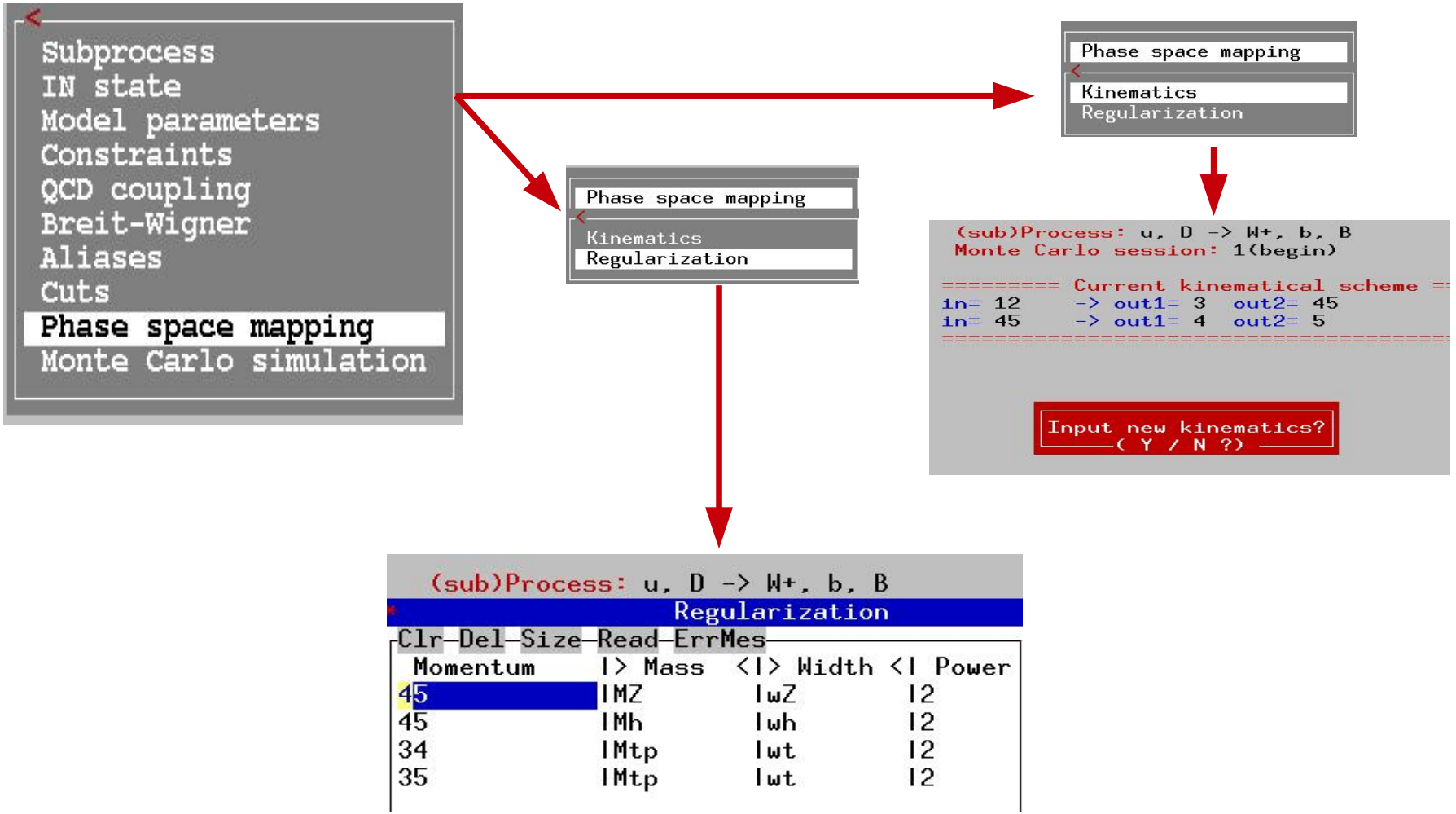
PgDn

Cuts 5

Clr Del Size Read ErrMes

Parameter	|> Min bound	<|> Max bound
T(b)	|20	|
T(B)	|20	|
N(b)	|-5	|5
N(B)	|-5	|5
J(b,B)	|0.5	|

phase-space mapping



integration over the phase space

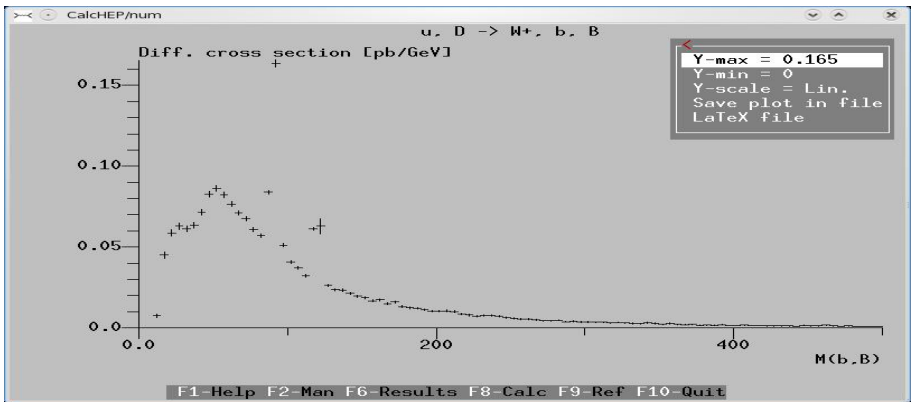
```
Subprocess
IN state
Model parameters
Constraints
QCD coupling
Breit-Wigner
Aliases
Cuts
Phase space mapping
Monte Carlo simulation
```

```
Monte Carlo simulation
nSess  = 5
nCalls = 10000
Set  Distributions
*Start integration
Display Distributions
Clear statistic
Freeze grid          OFF
Clear  grid
Event Cubes 10000
Generate Events
```

Distributions

Parameter_1	Min_1	Max_1	Parameter_2	Min_2	Max_2
T(b)	0	200			
T(B)	0	200			
N(b)	-5	5			
N(B)	-5	5			
M(b,B)	0	500			
M(W+,b)	0	500			
T(b)	0	500	M(b,B)	0	500

```
nSess  = 5
nCalls = 10000
Set  Distributions
*Start integration
Display Distributions
Clear statistic
Freeze grid          OFF
Clear  grid
Event Cubes 10000
Generate Events
```




```
(sub)Process: u, D -> W+, b, B
Monte Carlo session: 2(continue)
```

#IT	Cross section [pb]	Error %
6	9.5931E+00	7.10E-01
7	9.5686E+00	6.79E-01
8	9.5669E+00	6.82E-01
9	9.6892E+00	7.93E-01
10	9.6267E+00	7.51E-01
1	9.7757E+00	7.32E-01

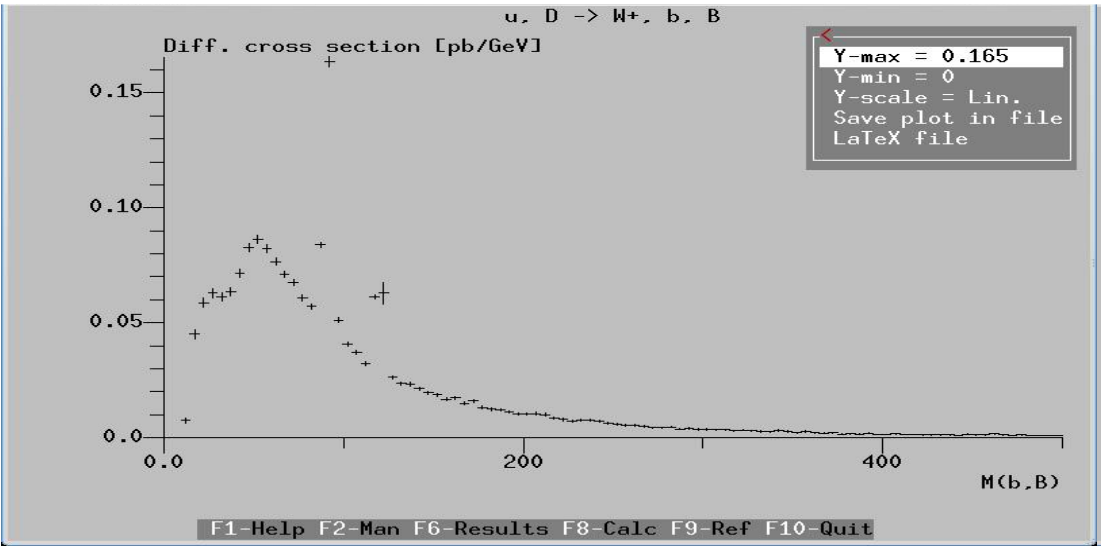
clear statistics.

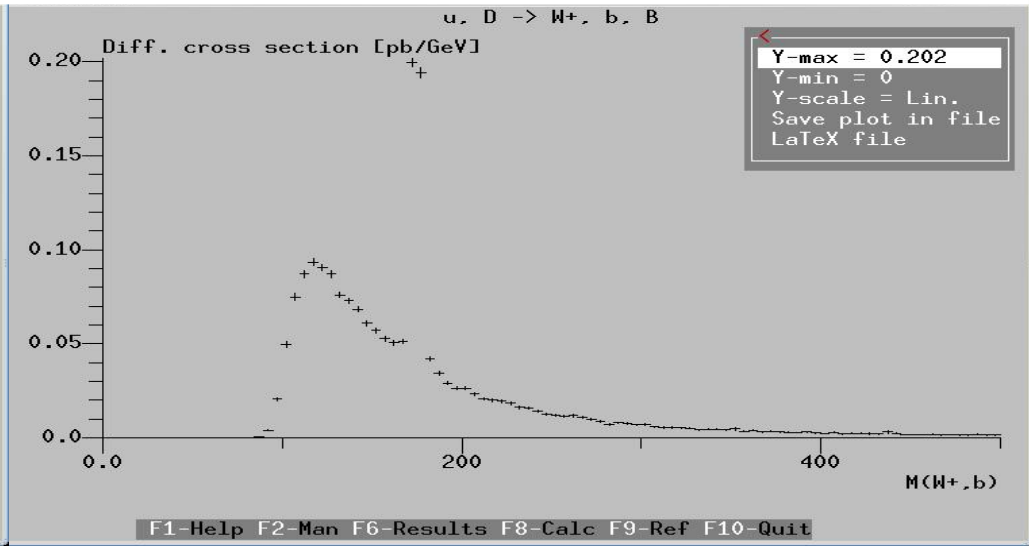
2	9.6557E+00	6.82E-01
3	9.7464E+00	1.38E+00
4	9.6845E+00	1.05E+00
5	9.7032E+00	7.68E-01
< >	9.7095E+00	3.74E-01

```
nSess  = 5
nCalls = 10000
Set  Distributions
*Start integration
Display Distributions
Clear statistic
Freeze grid          OFF
Clear  grid
Event Cubes 10000
Generate Events
```

The accuracy and the stability of the cross section indicate that you can trust your results

Resulting M_{bb} and M_{Wtb} kinematical distributions

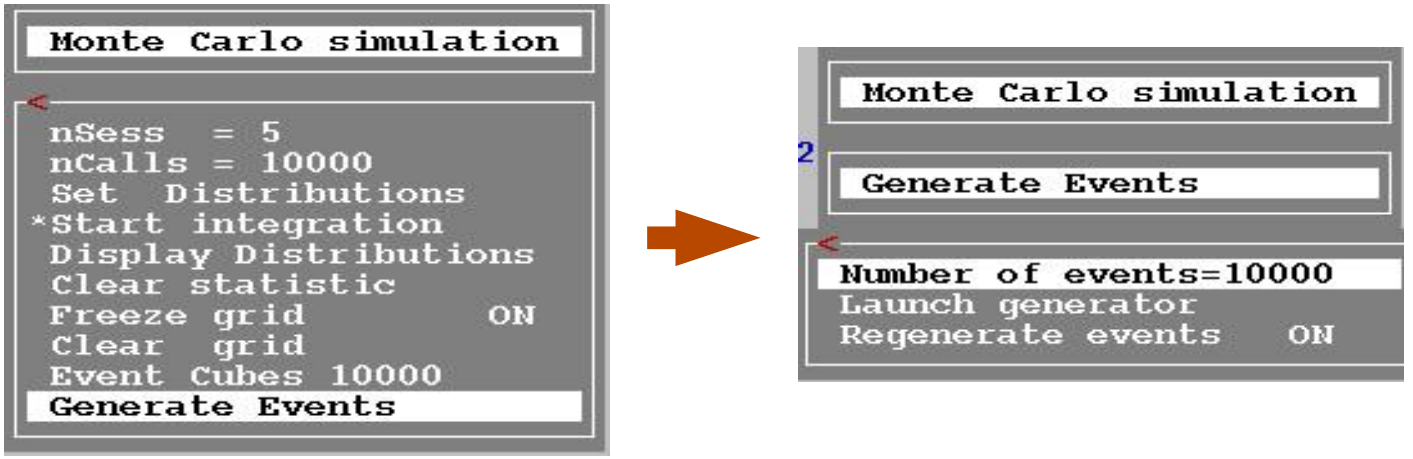


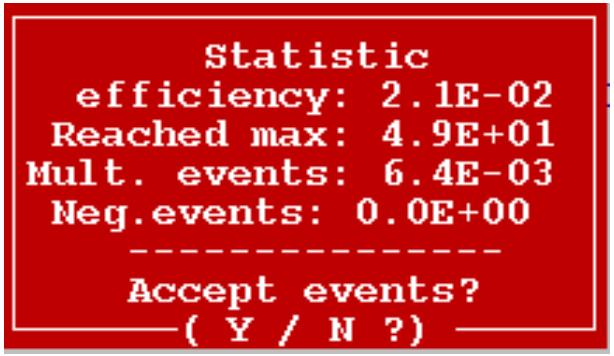


ex#5

1. Calculate WbB production rates at the LHC
for PT b-jet > 20 GeV, b-Jet separation > 0.5, -3< pseudorapidity < 3
2. Plot bb- and Wb invariant mass distributions for PT b-jet > 20 GeV and PT b-jet > 40 GeV

events generations





File with events in the native CalcHEP format

```
#Type 2 -> 3
#Initial_state
  P1_3=4.000000E+03  P2_3=-4.000000E+03
  StrFun1="PDT:cteq6m(proton)" 2212
  StrFun2="PDT:cteq6m(proton)" 2212
#PROCESS   2(u) -1(D) -> 24(W+) 5(b) -5(B)
#MASSES  0.0000000000E+00 0.0000000000E+00 8.0385000000E+01 3.2414139578E+00 3.2414139578E+00
#Cross_section(Width) 6.473084E+01
#Number_of_events        1000
#Events        P1_3 [Gev]         P2_3 [Gev]         P3_1 [Gev]         P3_2 [Gev]         P3_3 [Gev]         P4
       1   7.0828325272E+02 -3.8182148276E+00 -5.8685533663E+00  2.4810106784E+00  6.8128552155E+02  1.995
       1   1.5237718262E+02 -2.5952742306E+01  1.1734367441E+01 -2.1669699291E+01  5.6645397996E+01  4.499
       1   7.2370755716E+02 -3.3186893665E+00 -3.4449322581E+00 -5.1815667765E+00  5.8508268207E+02 -3.584
       1   2.6295673814E+02 -1.1370528114E+01  8.9463043464E+00 -3.4258266547E+00  2.2732569389E+02 -9.675
       1   5.7099697940E+02 -3.3943984194E+01  7.2879879961E+00 -2.3531627752E+01  1.9857446272E+01 -8.750
       1   3.6709401207E+02 -2.4124155464E+01 -4.8101350483E+00  6.6698730251E+01  2.0295672218E+02 -4.597
       1   3.7196555447E+01 -4.1553021555E+02 -3.1735918986E+00  2.8330641675E-01 -6.6745521993E+00  4.343
       1   4.0543944850E+01 -1.1104274125E+02 -8.2903700266E+00 -4.3292277920E+00 -9.0241583360E-01  6.562
       1   4.0084952687E+02 -1.0215920577E+01  1.1427574950E+01  2.6016502364E+00  3.8645254998E+02 -4.666
       1   2.2620009412E+01 -1.2387066011E+02 -5.0869818859E+00  1.1389105773E+01 -7.1200204784E+01  1.176
       1   7.2046251695E+02 -2.1091178466E+01 -1.4887347954E+01  8.1292985197E+01  5.8742582956E+02 -5.134
       1   6.8661185459E+01 -8.3534206530E+01 -5.5091602956E+00 -1.7099072377E+01  4.1559702536E+01  2.604
       1   1.5145483971E+03 -3.1164597600E+00 -7.8325298677E+00  3.6606202670E+01  1.2782056265E+03  1.074
```

GUI: full control of details of the process

scripts: automate calculation/generation/analysis

batch: does everything (sym,num,plots,…) in one run

Script example:

- **$CALCHEP/bin/subproc_cycle** ***nmax lumi***

  e.g.

  ```
  ../bin/subproc_cycle 1000 100000
  ```

  You should run it from results dir where the n_calchep binary is!

  Will evaluate cross section and generate events

- **$CALCHEP/bin/event_mixer Luminosity[1/fb] nevents event_dirs**

  mixes subprocesses and connects production and decay events

useful scripts for numerical session

see calchep_x.y.z/bin/ directory and README file!

- subproc_cycle ../bin/subproc_cycle 1000 100000
- sum_distr ../bin/sum_distr distr_2 distr_3 > distr_sum
- show_distr ../bin/show_distr distr_sum
- plot_view ../bin/plot_view < tab_1.txt
- events2tab
- lhe2tab
- gen_events
- name_cycle
- pcm_cycle
- par_scan
- event_mixer

ex#6
produce LHE file
and use lhe2tab
to produce
distributions from ex#5

Example of running subproc_cycle for SM model

```
hepul:~/calchep/work_demo/work/results> ../bin/subproc_cycle 1000 40
1000 events are requested
Number of enents limited by flux 40 [1/fb]

#Subprocess 1 ( d, U -> W-, b, B ) Cross section = 5.2061E+00 pb (1.94E+00%) , 1000 events
#Subprocess 2 ( D, u -> W+, b, B ) Cross section = 8.1961E+00 pb (1.15E+00%) , 1000 events
#Subprocess 3 ( u, D -> W+, b, B ) Cross section = 8.2515E+00 pb (8.99E-01%) , 1000 events
#Subprocess 4 ( U, d -> W-, b, B ) Cross section = 5.2065E+00 pb (1.39E+00%) , 1000 events
#Subprocess 5 ( s, C -> W-, b, B ) Cross section = 8.9595E-01 pb (8.10E-01%) , 1000 events
#Subprocess 6 ( S, c -> W+, b, B ) Cross section = 8.8594E-01 pb (7.34E-01%) , 1000 events
#Subprocess 7 ( c, S -> W+, b, B ) Cross section = 9.0529E-01 pb (7.46E-01%) , 1000 events
#Subprocess 8 ( C, s -> W-, b, B ) Cross section = 9.0196E-01 pb (5.65E-01%) , 1000 events

3.045E+01  -total cross section[pb]
3690       -maximum number of events
 1000 events are generated
Events in LHE format: events_9_16.lhe
Total Cross Section 3.046E+01 [pb] (5.685E-01%)
See details in directory 9_16
```

We need Events in LHE format to talk to MC generators!

- **bin/event_mixer** ***Luminosity[1/fb] nevents event_dirs***

mixes subprocesses and connects production and decay events

```
bin/event_mixer  10 1000 pp_wbb w_2x
9.327E+00  -total cross section[pb]
3265    -maximum number of events
```

- ***the output is*** **event_mixer.lhe** ***file***

```
<LesHouchesEvents version="1.0">
<!--
File generated with CalcHEP-PYTHIA interface
-->
<header>
<slha>
</slha>
</header>
<init>
  2212   2212   7.00000006860E+03   7.00000006860E+03     -1     -1     -1     -1      3      1
  1.16593335502E+01   0.00000000000E+00   1.00000000000E+00    1
</init>
<event>
 7     1    1.0000000E+00    2.8420000E+02  -1.0000000E+00  -1.0000000E+00
       -3    -1     0     0     0   501   0.00000000000E+00   0.00000000000E+00   1.54424456520E+02
        4    -1     0     0   500     0   0.00000000000E+00   0.00000000000E+00  -1.30792414700E+02
       24     2     1     2     0     0  -9.99292465447E+01  -1.63668803915E+01  -6.48692987742E+01
        5     1     1     2   500     0   7.34149473360E+01   2.15593961832E+01   4.23390519202E+01
       -5     1     1     2     0   501   2.65142992097E+01  -5.19251579179E+00   4.61622886720E+01
      -11     1     3     3     0     0  -7.19345413730E+01   7.47572186340E-01  -8.03452022142E+01
       12     1     3     3     0     0  -2.79947051718E+01  -1.71144525779E+01   1.54759034400E+01
</event>
```

Accessing all your results

- results are stored in “results” directory
- output files:
  - n_calchep — numerical module
  - prt_nn — protocol
  - distr_nn_mm — summed distributions
  - distr_nn — individual distribution
  - events_nn.txt — events file
  - list_prc.txt — list of processes
  - qnumbers — qnumbers – PYTHIA input with new prt definitions
  - session.dat — current session status – format is similar to prt_nn one
- for every new process the “results” directory is offered to be renamed or removed

protocol prt_*nn*

```
    CalcHEP kinematics module
 The session parameters:

#Subprocess 1 ( u, D -> W+, b, B )
#Session_number 1
#Initial_state  inP1=7.000000E+03  inP2=7.000000E+03
 Polarizations= { 0.000000E+00  0.000000E+00 }
  StrFun1="PDT:cteq6m(proton)" 2212
  StrFun2="PDT:cteq6m(proton)" 2212

#Physical_Parameters
    alfEMZ = 7.818060999999999E-03
    alfSMZ = 1.172000000000000E-01
.............................................
.............................................
#Cuts
*** Table ***
 Cuts
  Parameter   |> Min bound <|> Max bound <|
T(b)          |20            |
T(B)          |20            |
.............................................
.............................................
#Regularization
*** Table ***
 Regularization
 Momentum     |> Mass  <|> Width <| Power|
45            |MZ       |wZ       |2
45            |Mh       |wh       |2
.............................................
.............................................
#END
=====================================
 #IT   Cross section [pb]  Error %    nCall    chi**2
  1      2.0373E+00       3.30E+01    20000
  2      8.6164E+00       2.86E+01    20000
.............................................
.............................................
```

scripts for numerical session

- events2tab

  Parameters:
  1- name of variable,
  2- minimum limit,
  3- maximum limit,
  4- number of bins(<=300).
  File with events must be passed to input.
  ../bin/events2tab “T(b)” 1 100 200 < events_1.txt >tab.txt
  ../bin/tab_view < tab.txt

- name_cycle

  1: Name of parameter
  2: Initial value
  3: Step
  4: Number of steps
  ../bin/name_cycle Mh 100 10 11

scripts are used by calchep_batch interface – see below

the most general scan with par_scan

- Usage:
  $CALCHEP/bin/par_scan < data_file
- Data file structure:

```
# Comments following the '#' symbol
par_name_1  par_name_2 ...   par_name_N & fun_name_1  fun_name_2 ...
  val11       val12    ....    val1N
  val21       val12    ....    val1N
.........................
```

- where par_name_i present free parameters of the models. Among them one also can write momenta of incoming particles as momentum1 and momentum2.
- fun_name_i is the name of constrained parameter which will be presented in output file
- Output file has the same structure as input plus calculated numerical values for constrained parameters, and an additional column for evaluated cross section with statistical error
- If you are not interested in the prt_# files you can clean it using
  $CALCHEP/bin/par_scan clean < data_file

CalcHEP batch interface

CalcHEP batch interface: all results in one shot

```
Model:          SM(+hgg)
Model changed: False
Gauge:          Feynman
Process:    p,p->W,b,B
Decay:      W->le,n
########################################
Composite: p=u,U,d,D,s,S,c,C,b,B,G
Composite: W=W+,W-
Composite: le=e,E,m,M
Composite: n=ne,Ne,nm,Nm
Composite: jet=u,U,d,D,s,S,c,C,b,B,G
########################################
pdf1:       PDT:cteq6l1(proton)
pdf2:       PDT:cteq6l1(proton)
p1:         6500
p2:         6500
########################################
Run parameter: Mh
Run begin:     120
Run step size: 5
Run n steps:   3
########################################
alpha Q :             M45
########################################
Cut parameter:    M(b,B)
Cut invert:       False
Cut min:          100
Cut max:
```

```
Kinematics :       12 -> 3, 45
Kinematics :       45 -> 4 , 5
Regularization momentum:1: 45
Regularization mass:1:     Mh
Regularization width:1:    wh
Regularization power:1:    2
########################################
Dist parameter:    M(b,B)
Dist min:          100
Dist max:          200
Dist n bins:       100
Dist title:        p,p->W,b,B
Dist x-title:      M(b,B) (GeV)

Dist parameter:    M(W,jet)
Dist min:          100
Dist max:          200
Dist n bins:       100
Dist title:        p,p->W,b,B
Dist x-title:      M(W,jet) (GeV)
########################################
Number of events (per run step): 10000
Filename:                        pp_Wbb
########################################
Parallelization method:      local
Max number of nodes:   8
Max number of processes per node: 1
########################################
nSess_1:   5
nCalls_1:  100000
nSess_2:   5
nCalls_2:  100000
```

CalcHEP batch interface: running and monitoring

$CALCHEP= path to calchep installation

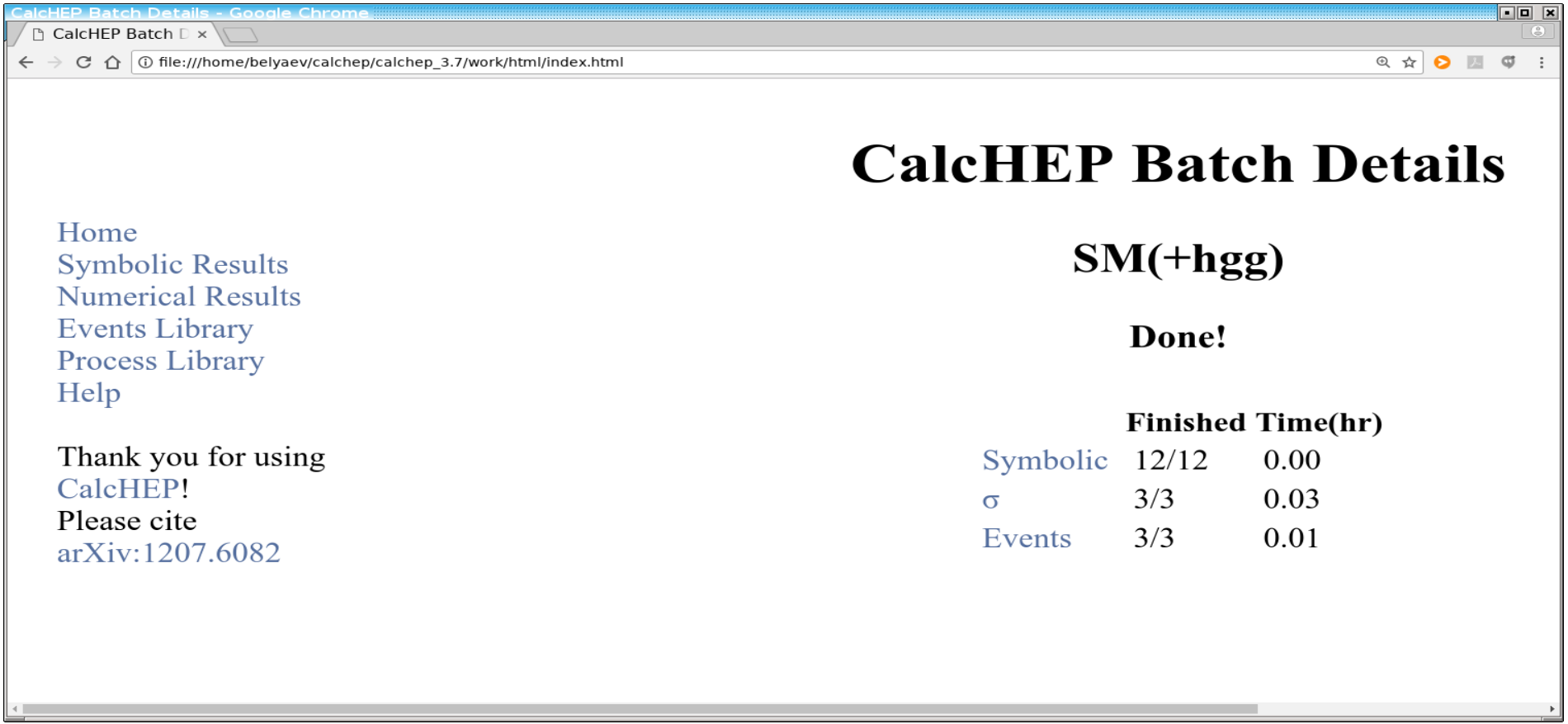
cd $CALCHEP/work

cp ../utile/batch_file .

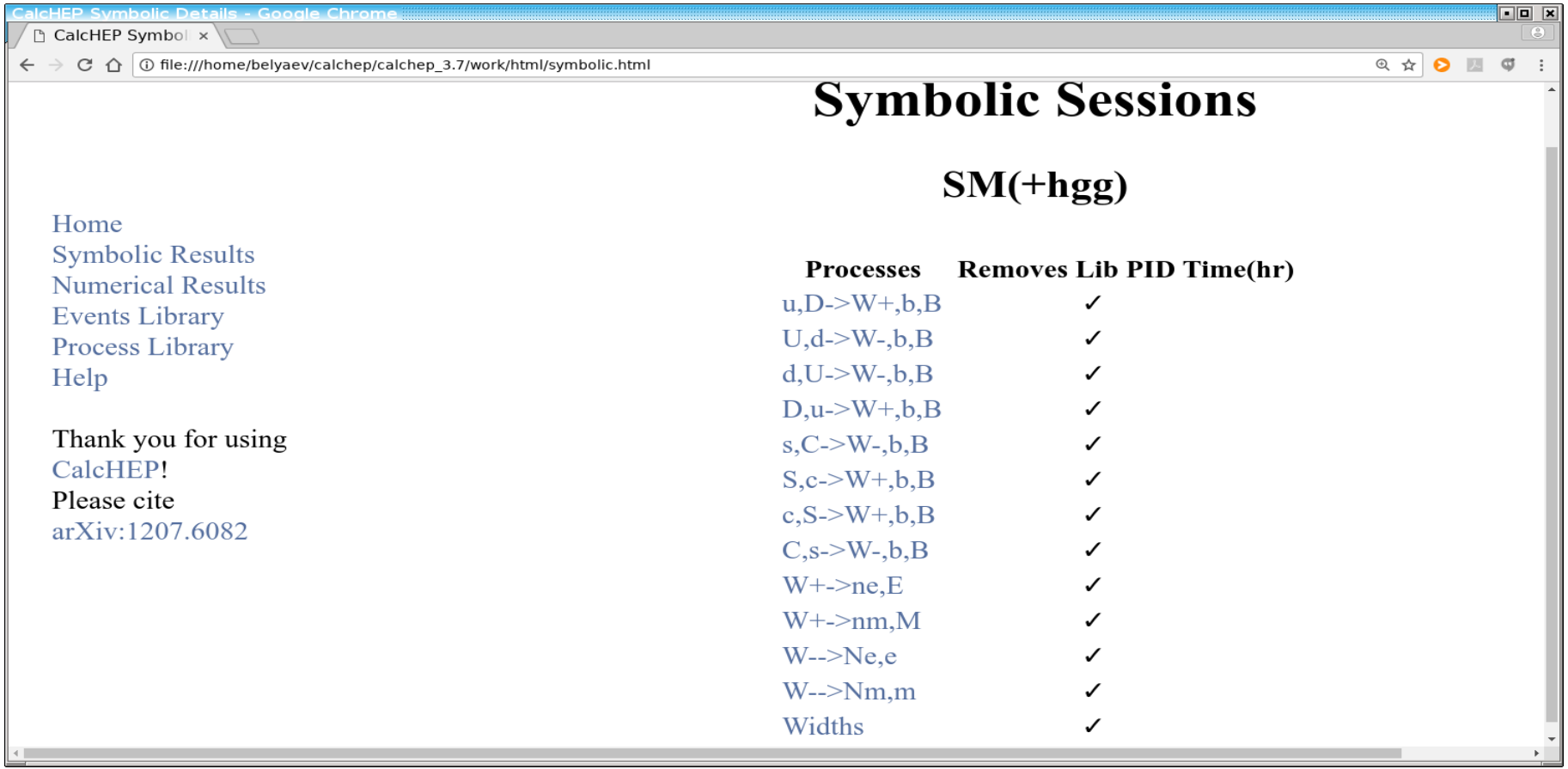
./calchep_batch batch_file

```
CALCHEP= /home/belyaev/calchep/calchep_x.y.z

calchep_batch version x.yz

Processing batch:
Progress information can be found in the html directory.
Simply open the following link in your browser:
file:///.../calchep_x.y.z/work/html/index.html

You can also view textual progress reports in
.../calchep_x.y.z/work/html/  and the other .txt files in the html directory.
Events will be stored in the batch_results directory.
```

CalcHEP batch interface: running and monitoring



CalcHEP batch interface: running and monitoring



CalcHEP batch interface: running and monitoring

CalcHEP Numerical Details - Google Chrome

CalcHEP Numeri ×

file:///home/belyaev/calchep/calchep_3.7/work/html/numerical.html

Numerical Sessions

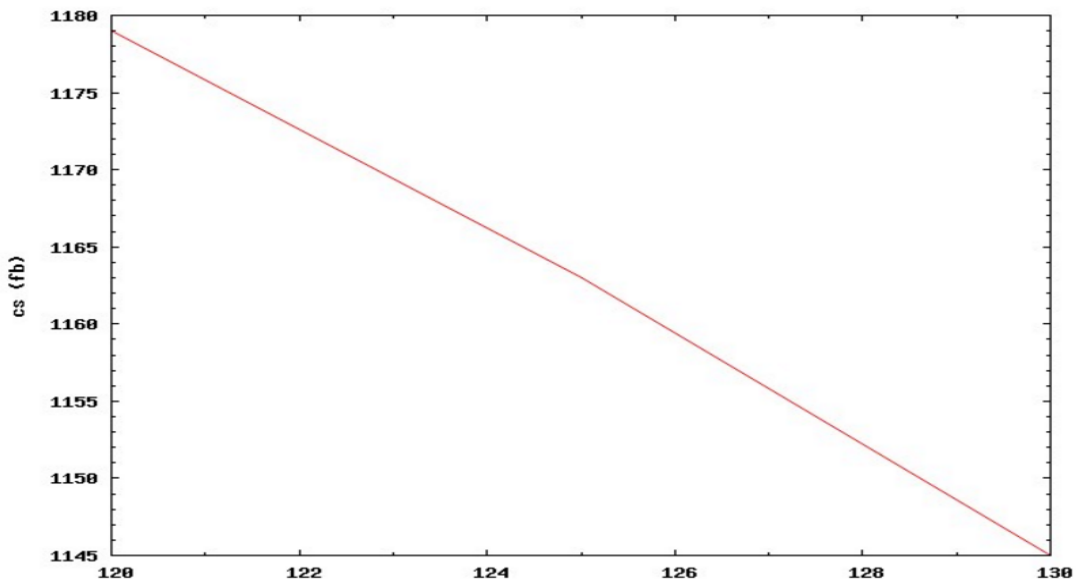
SM(+hgg)

Done!

Home
Symbolic Results
Numerical Results
Events Library
Process Library
Help

Thank you for using CalcHEP!
Please cite
arXiv:1207.6082

Scans	σ (fb)	Running	Finished	Time (hr)	N events
Mh=120	1179	0/13	13/13	0.01	1000
Mh=125	1163	0/13	13/13	0.01	1000
Mh=130	1145	0/13	13/13	0.01	1000
				0.03	



CalcHEP batch interface: running and monitoring

Home
Symbolic Results
Numerical Results
Events Library
Process Library
Help

Thank you for using CalcHEP!
Please cite
arXiv:1207.6082

Numerical Sessions

SM(+hgg)

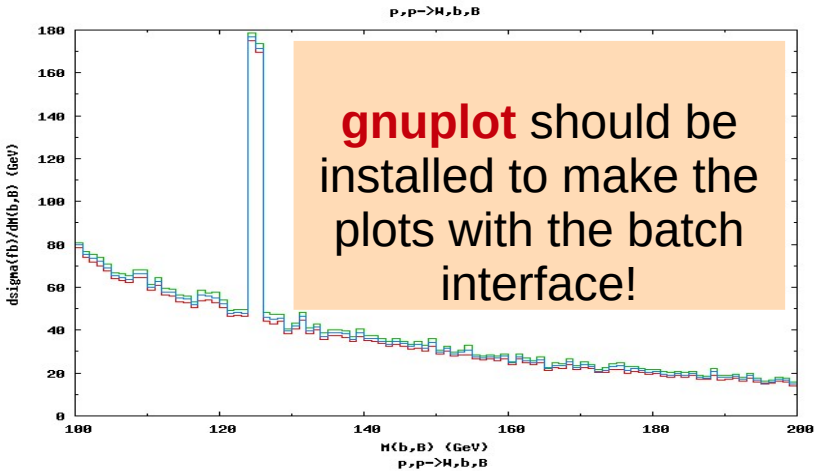
Done!

Processes	σ (fb)	Δσ (%)	PID	Time (hr)	N events	Details
u,D->W+,b,B	1552.8	0.8	28872	0.00	383/382	prt_1 session.dat
U,d->W-,b,B	829.4	0.52	28878	0.00	220/219	prt_1 session.dat
d,U->W-,b,B	837.46	1.1	28885	0.00	221/220	prt_1 session.dat
D,u->W+,b,B	1558.3	0.51	29100	0.00	384/383	prt_1 session.dat
s,C->W-,b,B	109.55	0.54	29104	0.00	42/41	prt_1 session.dat
S,c->W+,b,B	108.79	0.44	29109	0.00	41/40	prt_1 session.dat
c,S->W+,b,B	108.88	0.41	29116	0.00	41/40	prt_1 session.dat
C,s->W-,b,B	109.6	0.43	29123	0.00	42/41	prt_1 session.dat
Total	5214.8	0.34				

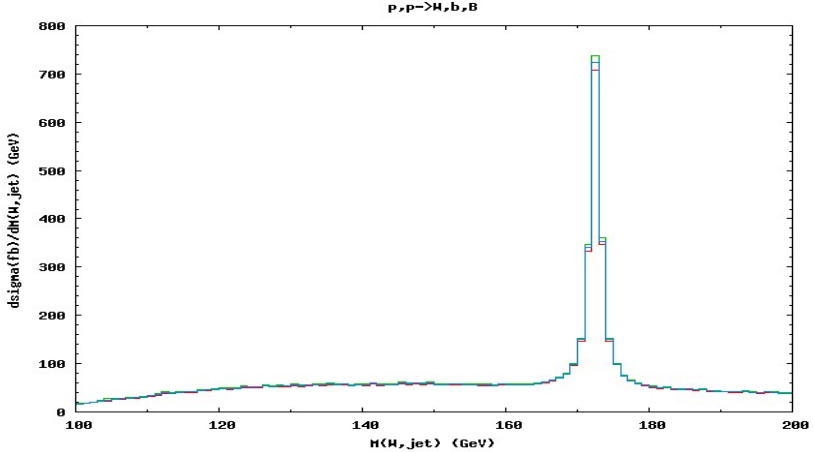
Decays	Γ (GeV)	ΔΓ (%)	PID	Time (hr)	N events	Details
W+->ne,E	0.23293	0	29129	0.00	5099/5100	prt_1 session.dat
W+->nm,M	0.23293	0	29135	0.00	5099/5100	prt_1 session.dat
W-->Ne,e	0.23293	0	29142	0.00	5099/5100	prt_1 session.dat
W-->Nm,m	0.23293	0	29324	0.00	5099/5100	prt_1 session.dat

Widths		PID	Time (hr)		Details
Widths		29328	0.00		session.dat
Total	1163		0.01	1000/1000	

Distributions



gnuplot should be installed to make the plots with the batch interface!



ex#7: *using calchep_batch evaluate complete cross section for pp-> Wbb process with the same cuts as for ex#5*

CalcHEP batch results

- results are located in **batch_results** folder
- ***.lhe.gz** : LHE event files
- ***.jpg** : figures
- ***.distr** : files with distributions which cab be used to re-produce plots using **$CALCHEP/bin/show_distr**
- ***.tgz** : zipped html folder with all numerical details, .txt and .html files of the batch run

CalcHEP batch interface: some additional features/tricks

- see https://answers.launchpad.net/calchep for many “tricky” questions/answers
- scanning over the collider energy

  Run parameter: rtS
  Run begin: 7
  Run step size: 1
  Run n steps: 2

  p1: 1000*rtS/2
  p2: 1000*rtS/2
  rtS here is some “fake” parameter

- you can use “fake” parameter only if you define it as a loop parameter
  - It can be used in the cut statement (assigning cut to the symbol)
  - It can be assigned to the parameter model – this way you can handle a complicated scan

CalcHEP batch interface: scanning with ‘fake’ parameters

an example:

Calculating SM background cross section for process pp→e+e- across invariant mass range of 500GeV-3TeV, with cut of +/-400GeV around invariant mass.

```
########################################
#  Run Info                            #
#                                      #
#  Masses and Energies are in GeV      #
#  More than one run can be specified  #
#  at the same time.                   #
########################################
Run parameter: MASS
Run begin:     500
Run step size: 500
Run n steps:   6
```

```
########################################
#  Cut Info                            #
#                                      #
# Must be in terms of the (production mode)
#     final state particles.           #
#     :n: specifies which process.     #
#     : means to apply to all processes.
########################################

Cut parameter:    M(e,E)
Cut invert:       False
Cut min:          MASS-400
Cut max:          MASS+400
```

CalcHEP interface to MC generators via Events in the LHE format