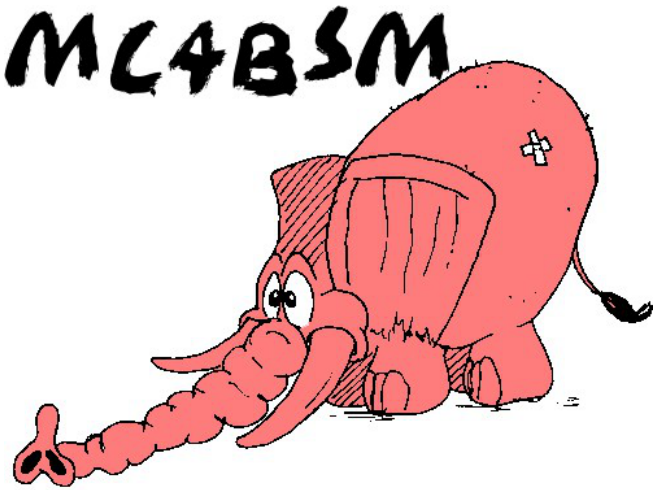
CalcHEP

a practical introduction

Alexander Belyaev

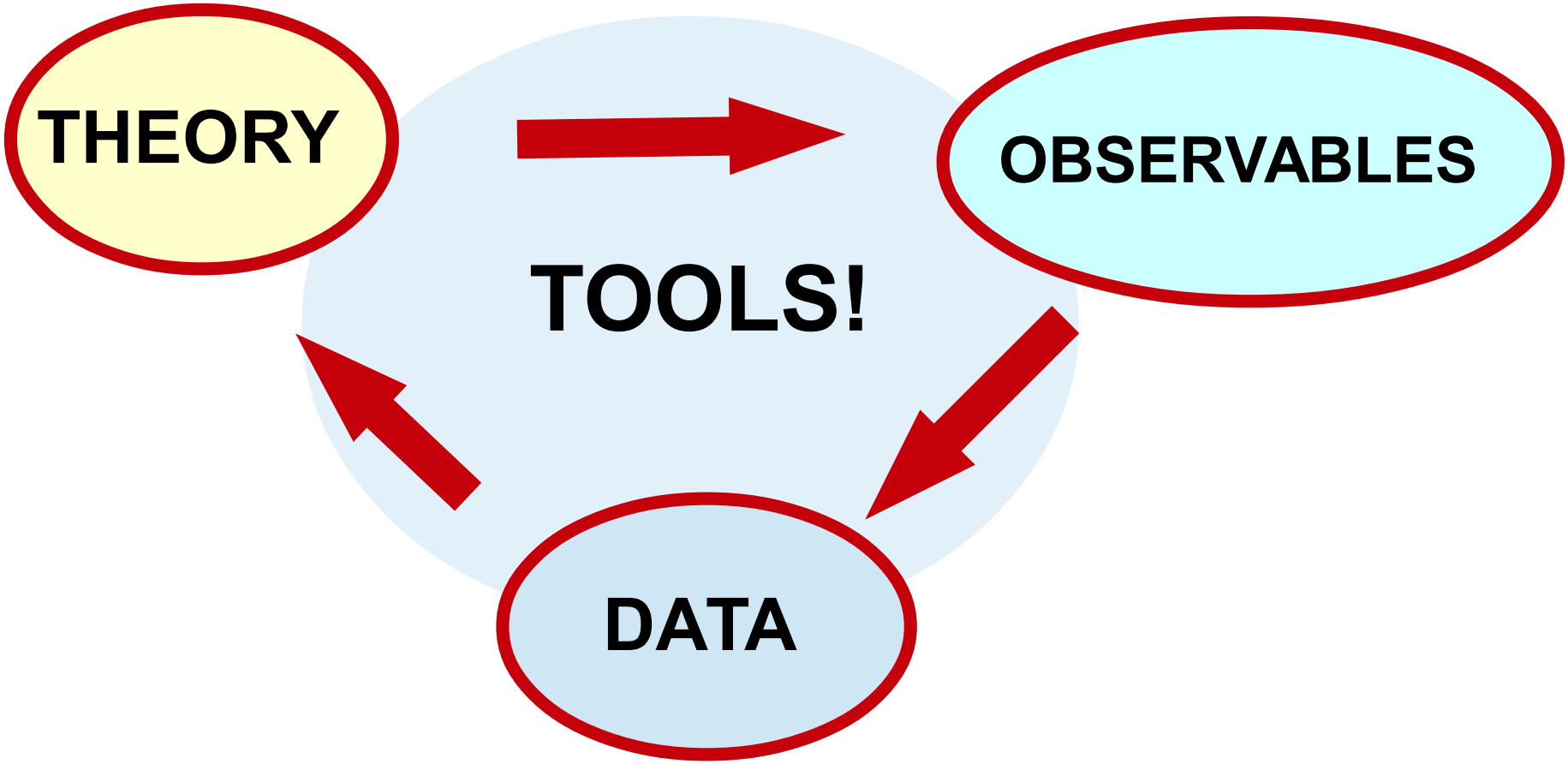
Southampton University &
Rutherford Appleton Laboratory

CalcHEP authors:
Alexander Pukhov, Alexander Belyaev, Neil Christensen

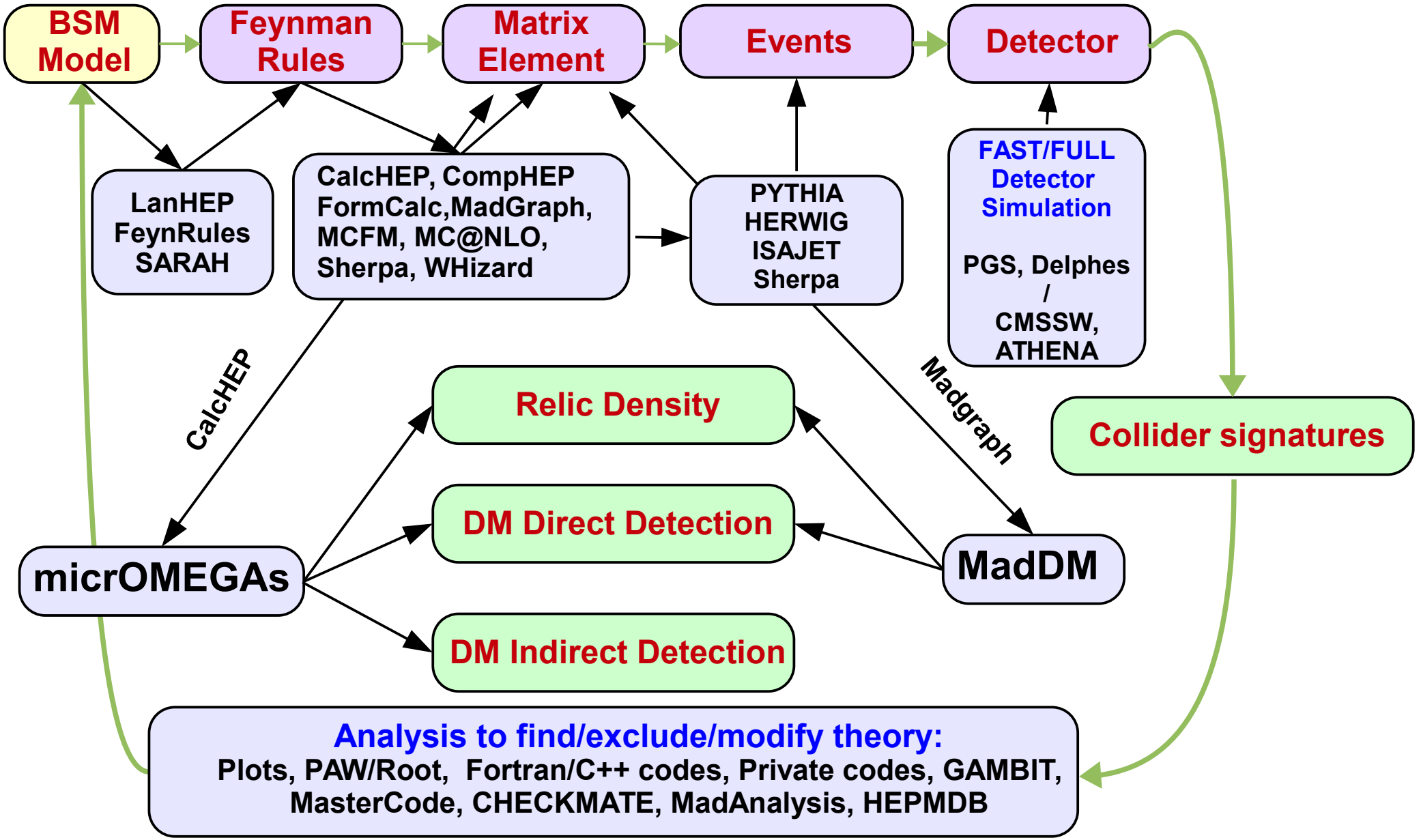


IPPP Durham, 19th of April 2018

theory ↔ data requires observables to be compared with data and we need TOOLS to do this!



Tools for theory → observables link



CalcHEP

Calculator for High Energy Physics

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

CalcHEP

Calculator for High Energy Physics

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

- **Authors**

  Alexander Pukhov, AB, Neil Christensen

  (AB and Neil Christensen have joined the project in 2009)
  http://theory.npi.msu.su/~pukhov/calchep.html

- **Idea**

  The effective study of HEP phenomenology passing at high level of automation from your favorite model to physical observables such as decay width, branching ratios, cross sections kinematic distributions, parton-level events, ...

CalcHEP

Calculator for High Energy Physics

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

- **Authors**

  Alexander Pukhov, AB, Neil Christensen

  (AB and Neil Christensen have joined the project in 2009)
  http://theory.npi.msu.su/~pukhov/calchep.html

- **Idea**

  The effective study of HEP phenomenology passing at high level of automation from your favorite model to physical observables such as decay width, branching ratios, cross sections kinematic distributions, parton-level events, ...

- **Analogous packages (matrix element generators)**

  - **CompHEP (Boos et al)**
  - **MadGraph/MadEvent (Maltoni, Stelzer et al)**
  - **Grace/Helas (Fujimoto et al)**
  - **FeynArts/FeynCalc/FormCalc (Hahn et al)**
  - **WHIZARD,O'mega (Moretti, Ohl, Reuter)**
  - **Sherpa (Krauss et al)**

General Features/Limitations of CalcHEP

- Can evaluate any decay and scattering processes within any (user defined) model!

General Features/Limitations of CalcHEP

- Can evaluate any decay and scattering processes within any (user defined) model!
- Tree-level processes

General 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 the final particles – spin averaged amplitude

General 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 the final 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

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details**
  (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details**
  (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **Output of symbolic results** (Mathematica, REDUCE, FORM formats)

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details**
  (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **Output of symbolic results** (Mathematica, REDUCE, FORM formats)
- **Calculates particle widths 'on the fly'**

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details**
  (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **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)

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details** (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **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)
- **Powerful batch interface:** connects production and decay processes, allows to perform multidimensional scan of the the parameter space and produce LHE files in one run

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details** (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **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)
- **Powerful batch interface:** connects production and decay processes, allows to perform multidimensional scan of the the parameter space and produce LHE files in one run
- **Connected to HEPMD** (High Energy Physics Model Database) **with HPC cluster behind the web interface**

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details**
  (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **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)
- **Powerful batch interface:** connects production and decay processes, allows to perform multidimensional scan of the the parameter space and produce LHE files in one run
- **Connected to HEPMD** (High Energy Physics Model Database) **with HPC cluster behind the web interface**
- **Designed to study physics and present and future colliders:**
  LHA PDF for LHC and ISR+Beamstrahlung for ILC

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details** (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **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)
- **Powerful batch interface:** connects production and decay processes, allows to perform multidimensional scan of the the parameter space and produce LHE files in one run
- **Connected to HEPMD** (High Energy Physics Model Database) **with HPC cluster behind the web interface**
- **Designed to study physics and present and future colliders:** LHA PDF for LHC and ISR+Beamstrahlung for ILC
- **Modular structure allows to use it as a matrix element generator in separate codes and packages** (e.g.GAMBIT)

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details** (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **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)
- **Powerful batch interface:** connects production and decay processes, allows to perform multidimensional scan of the the parameter space and produce LHE files in one run
- **Connected to HEPMD** (High Energy Physics Model Database) **with HPC cluster behind the web interface**
- **Designed to study physics and present and future colliders:** LHA PDF for LHC and ISR+Beamstrahlung for ILC
- **Modular structure allows to use it as a matrix element generator in separate codes and packages** (e.g.GAMBIT)
- **Connected to launchpad – see many questions and aswers** https://answers.launchpad.net/calchep

The Highlights of CalcHEP

- **Convenient graphical interface – to understand process in details** (e.g. one can select diagrams at squared level to study interference, plot dependent parameters, etc)
- **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)
- **Powerful batch interface:** connects production and decay processes, allows to perform multidimensional scan of the the parameter space and produce LHE files in one run
- **Connected to HEPMD** (High Energy Physics Model Database) **with HPC cluster behind the web interface**
- **Designed to study physics and present and future colliders:** LHA PDF for LHC and ISR+Beamstrahlung for ILC
- **Modular structure allows to use it as a matrix element generator in separate codes and packages** (e.g.GAMBIT)
- **Connected to launchpad – see many questions and aswers** https://answers.launchpad.net/calchep
- **Has different modules for user modfifications:** user-defined cuts, user form factor etc.

See tutorial for most of them

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 News&Bugs 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)

Code download.

License GPL-3

Installation Current version 3.7(updated 16.04.2018) New Options Archive

---

Models:

MSSM_10.14(15.10.2014) NMSSM_8.15(25.08.2015) CPVMSSM_10.14(16.10.2014) SUSY models By A.Semenov LeptoQuarks 5DSM 6DSM

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*

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 News&Bugs 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 3.7(updated 16.04.2018) New Options Archiv

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
5DSM 6DSM
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

Quick start with CalcHEP: practical notes on the installation

- Download code, read manual and compile
  http://theory.npi.msu.su/~pukhov/calchep.html

  wget https://theory.sinp.msu.ru/~pukhov/CALCHEP/calchep_3.7.tgz

  tar -zxvf calchmep_3.7.tgz

  cd calchep_3.7

  make

  cd work

  (*or your own work directory* ./mkWORKdir ../calc_work)
- You need to have only c compiler and X11 sources
- Supported operating system

  Linux, IRIX, IRIX64, HP-UX, OSF1, SunOS, Darwin, CYGWIN

  (see *getFlags* file)

ex#1: Install CalcHEP

Starting CalcHEP

- ***Files in the work folder:***

bin -> ..... /calchep_3.**x.y**/bin
calchep
calchep_batch
calchep.ini
models/
results/
tmp/

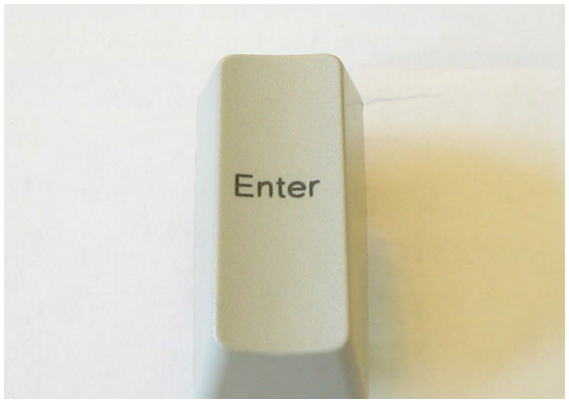
- ***Start:***

./calchep

Starting CalcHEP

```
CalcHEP_3.7/symb
    CalcHEP - a package for Calculation in High Energy Physics
            Version 3.7: Last correction  April  16, 2018


  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
```

Principle KEYS for CalcHEPs GUI



Enter menu selection (forward)

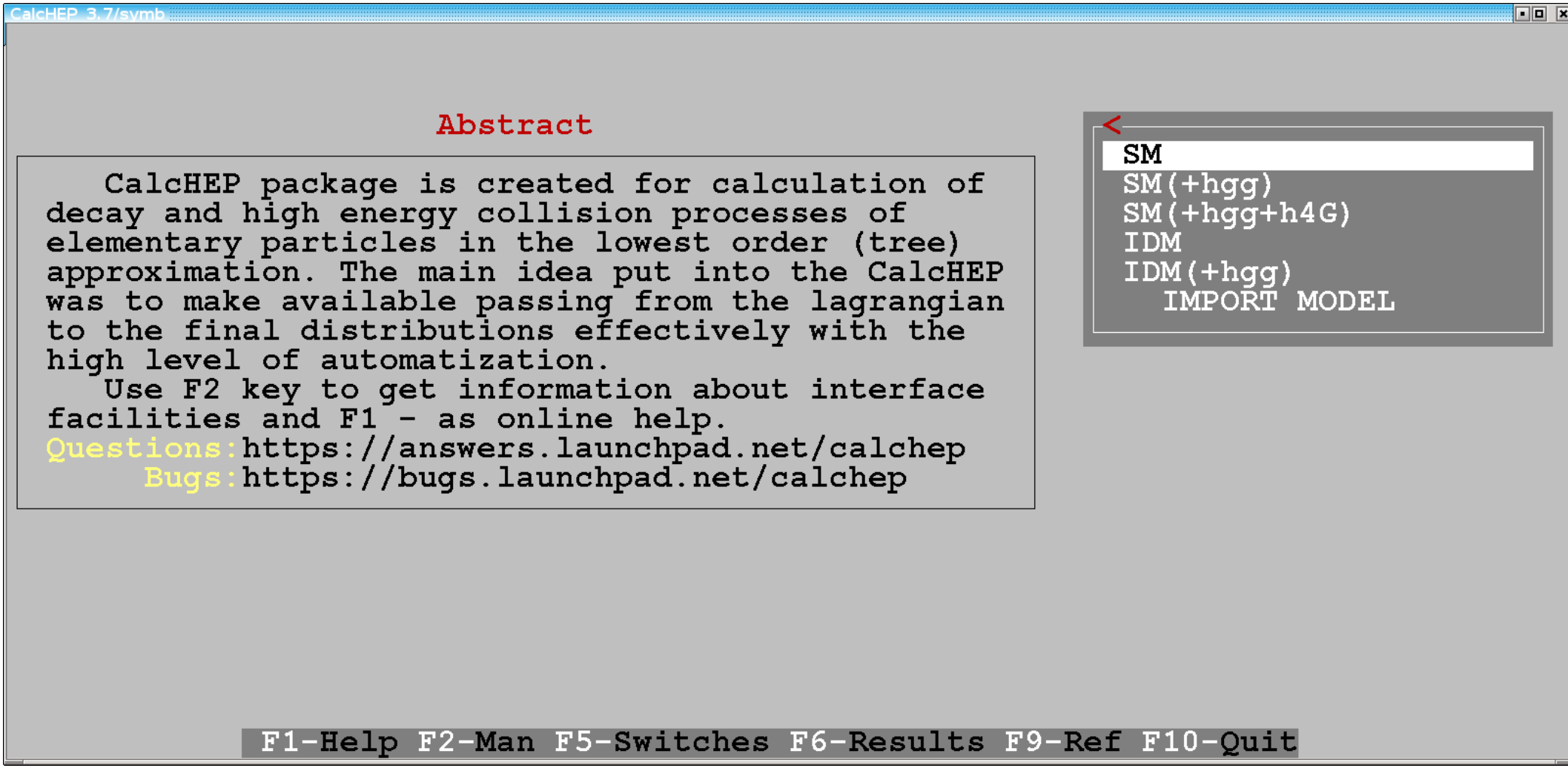
Exit menu selection (back)

Help!

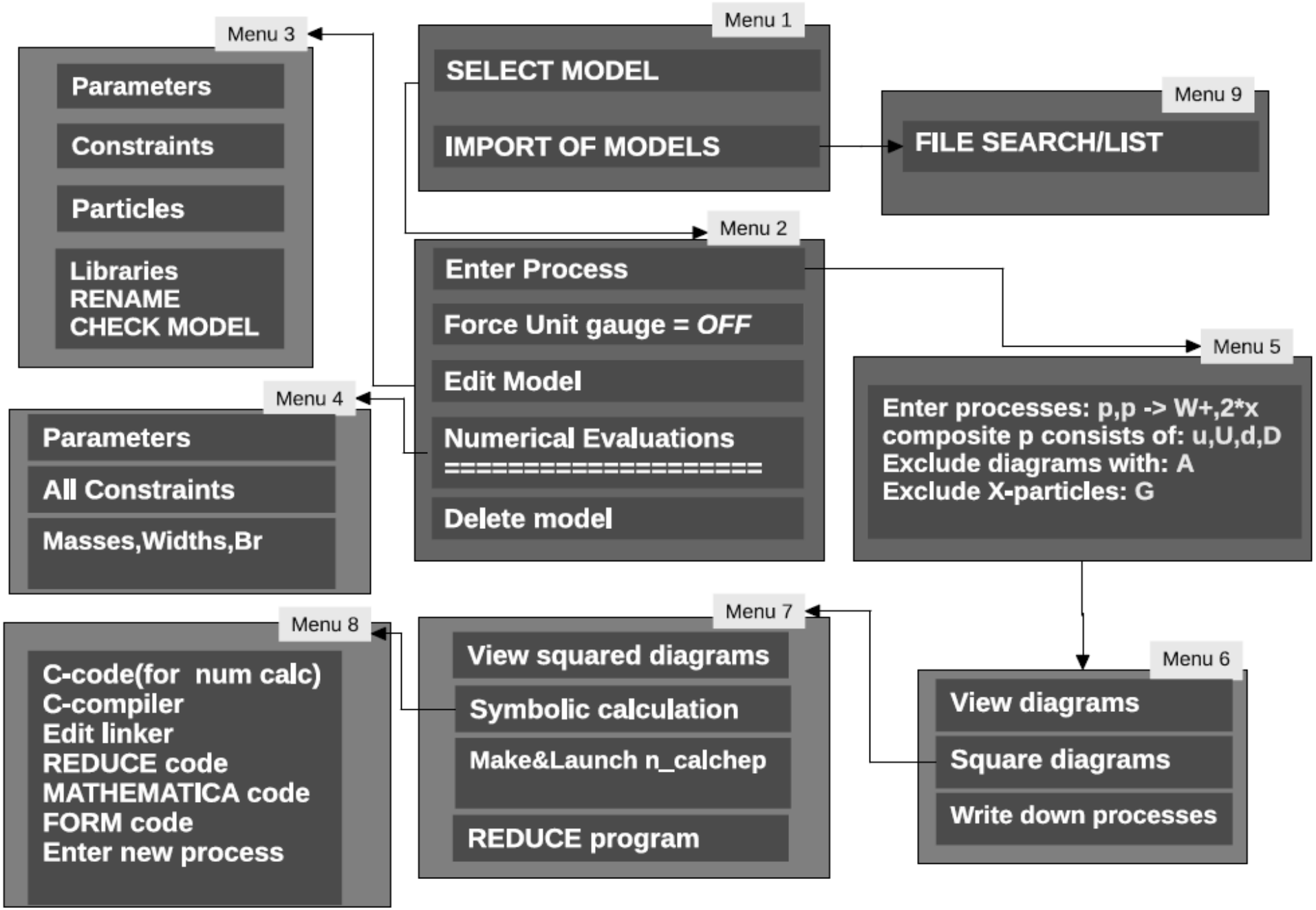
CalcHEP structure/modes

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

Starting CalcHEP

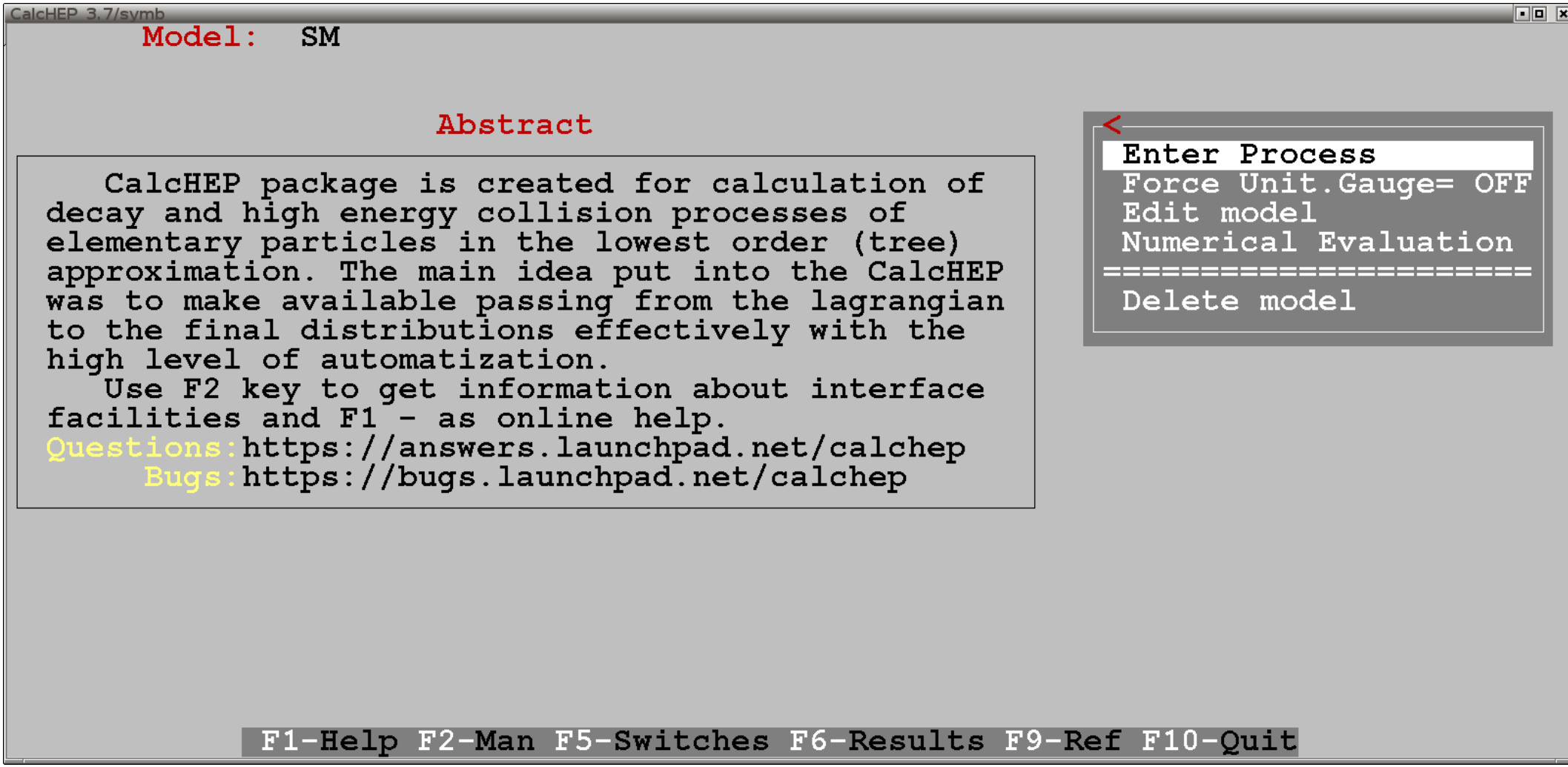


Symbolic part



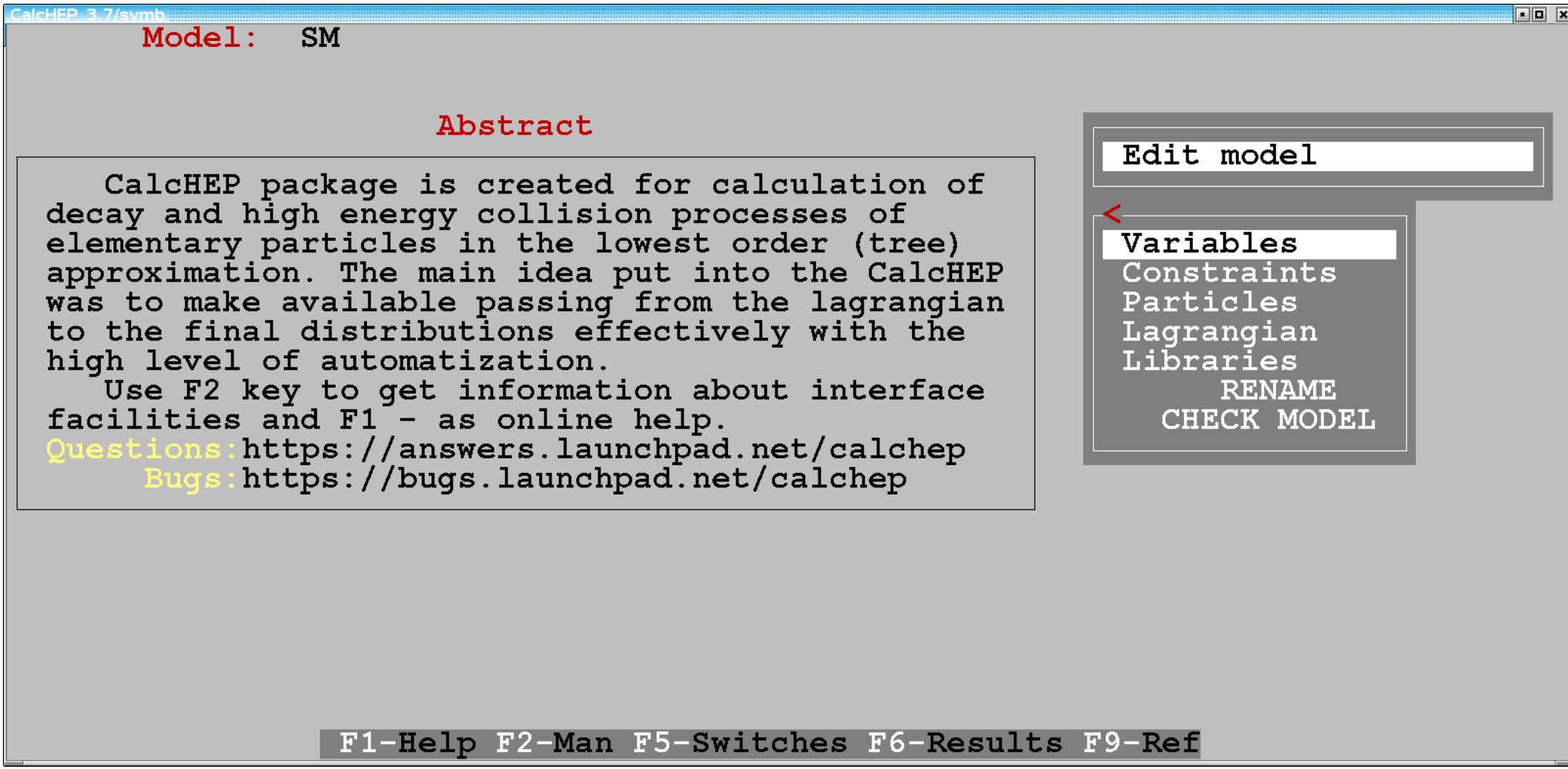
Model choice and Process input

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)

CalcHEP_3.6.28/symb

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

CalcHEP_3.6.28/symb

Particles

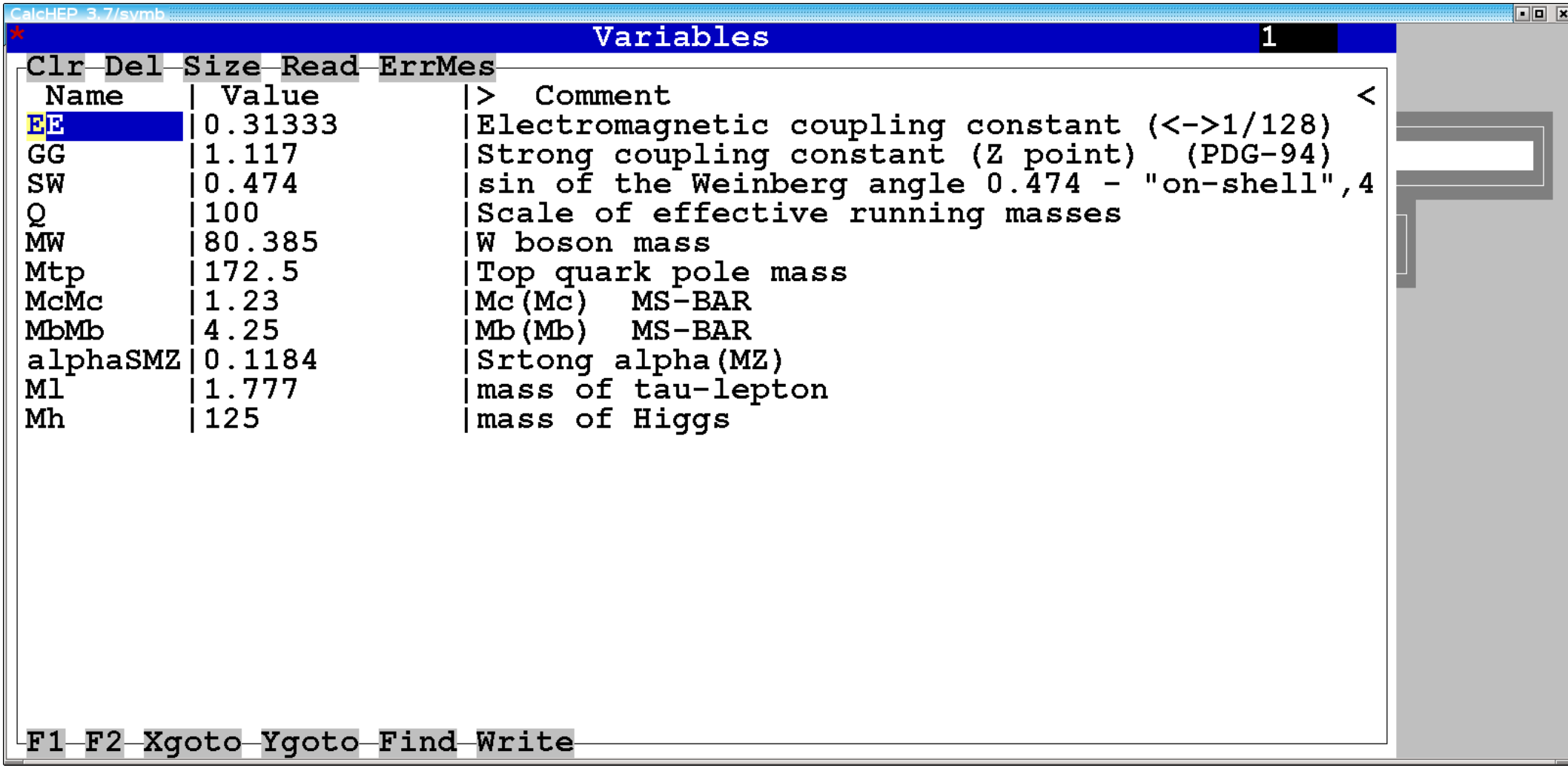
Clr Del Size Read ErrMes

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: vars*xx*.mdl



Dependent parameters(constraints): funcxx.mdl

CalcHEP_3.7/symb

Constraints

Clr Del Size Read ErrMes

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

F1 F2 Xgoto Ygoto Find Write

Feynman rules: lgrng*xx*.mdl

CalcHEP 3.7/symb

Lagrangian 1

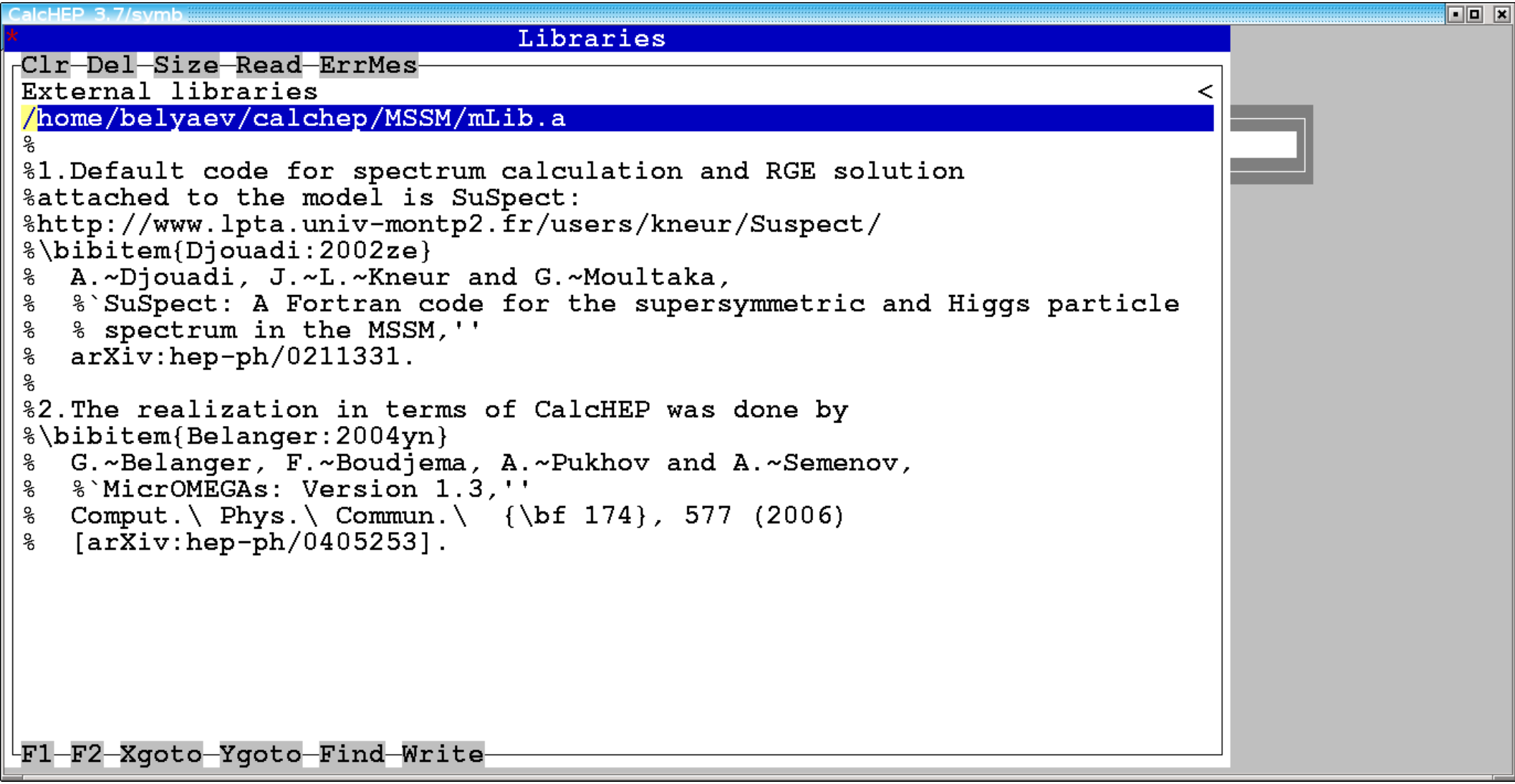
Clr Del Size Read ErrMes

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)

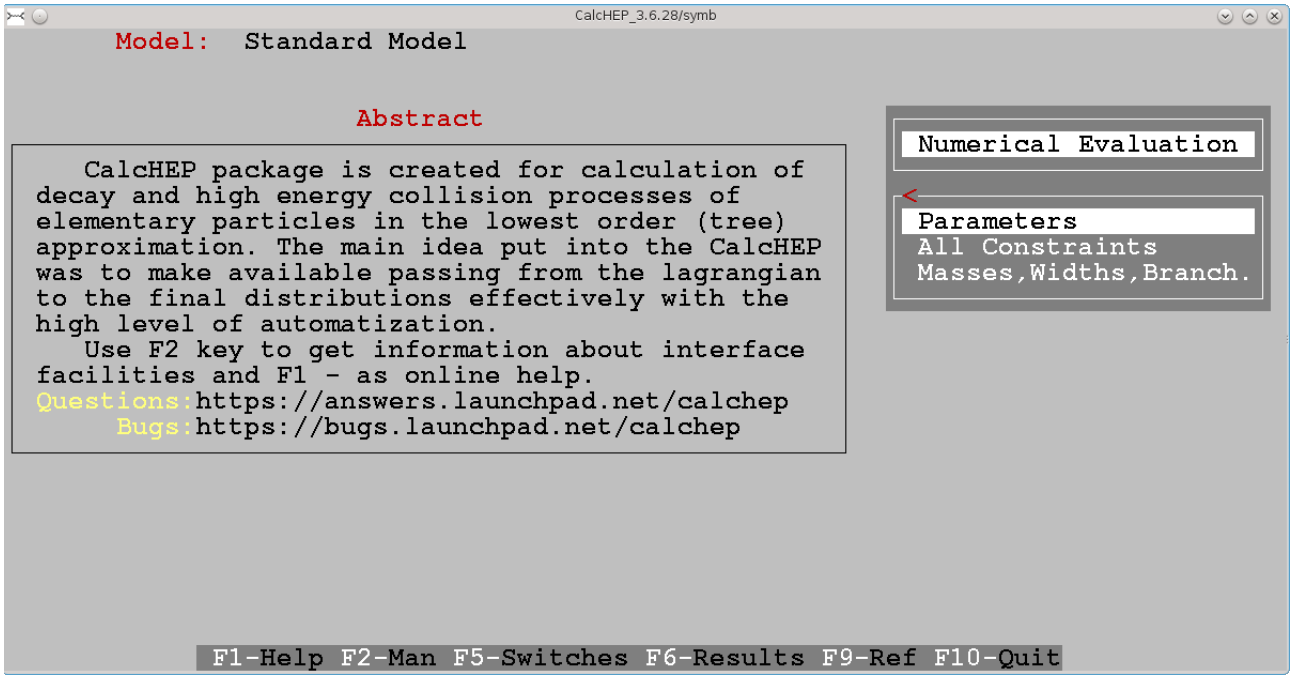
F1 F2 Xgoto Ygoto Find Write

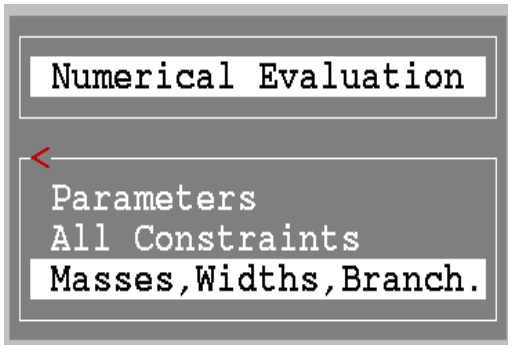
External Libraries: extlib*xx*.mdl

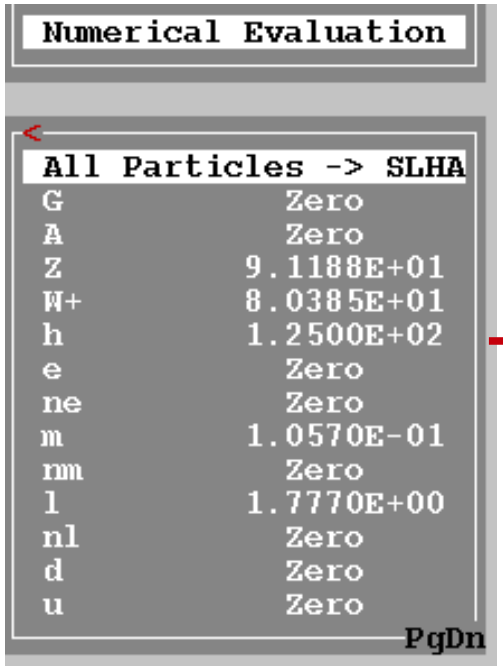
Typically is empty for simple models but can be used for any library which helps to build complicated model. E.g. mass spectra calculator for SUSY (involving RGE solutions etc)



Numerical evaluation of masses & branchings







ex#2: 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**

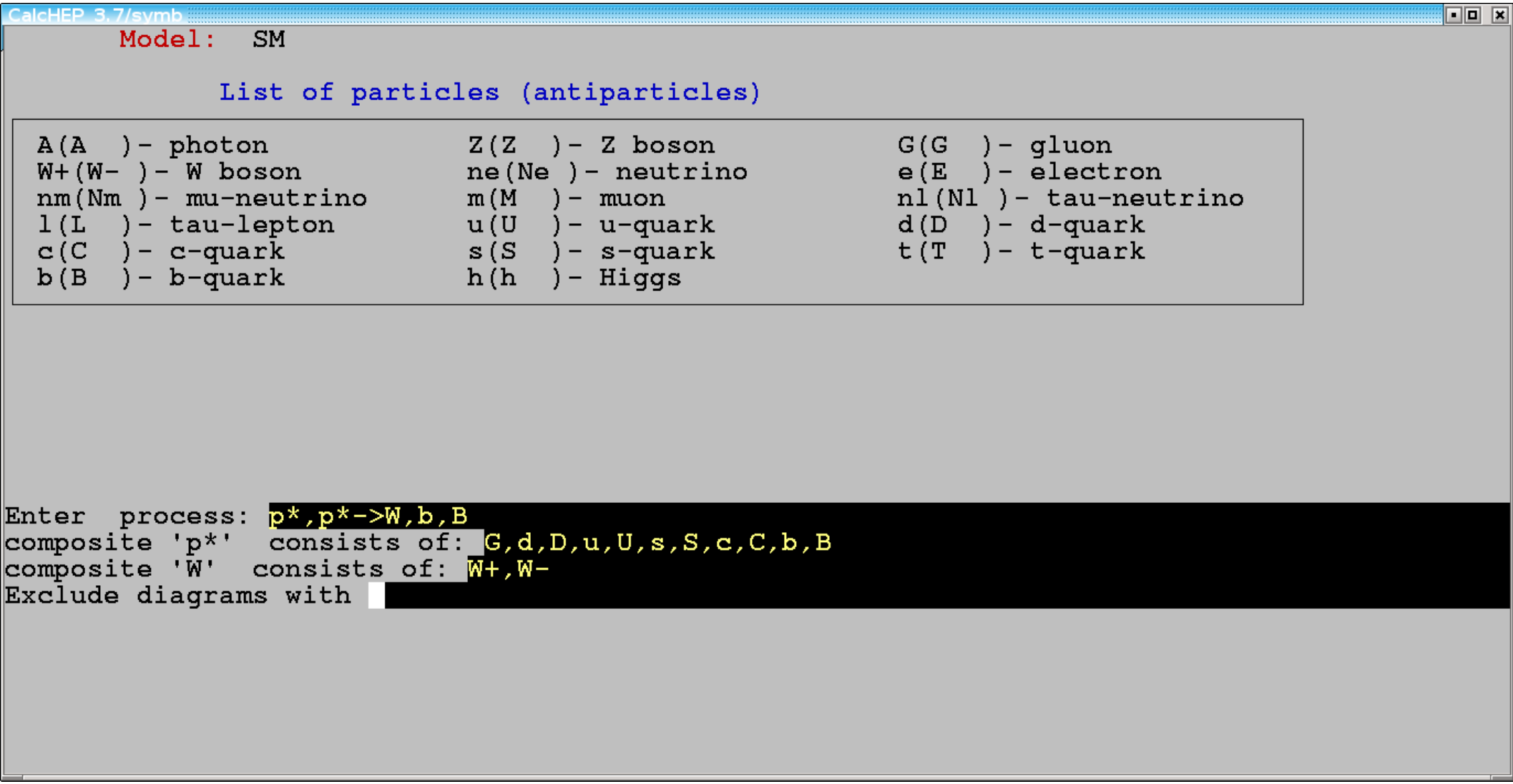
  'H -> 2*x'

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

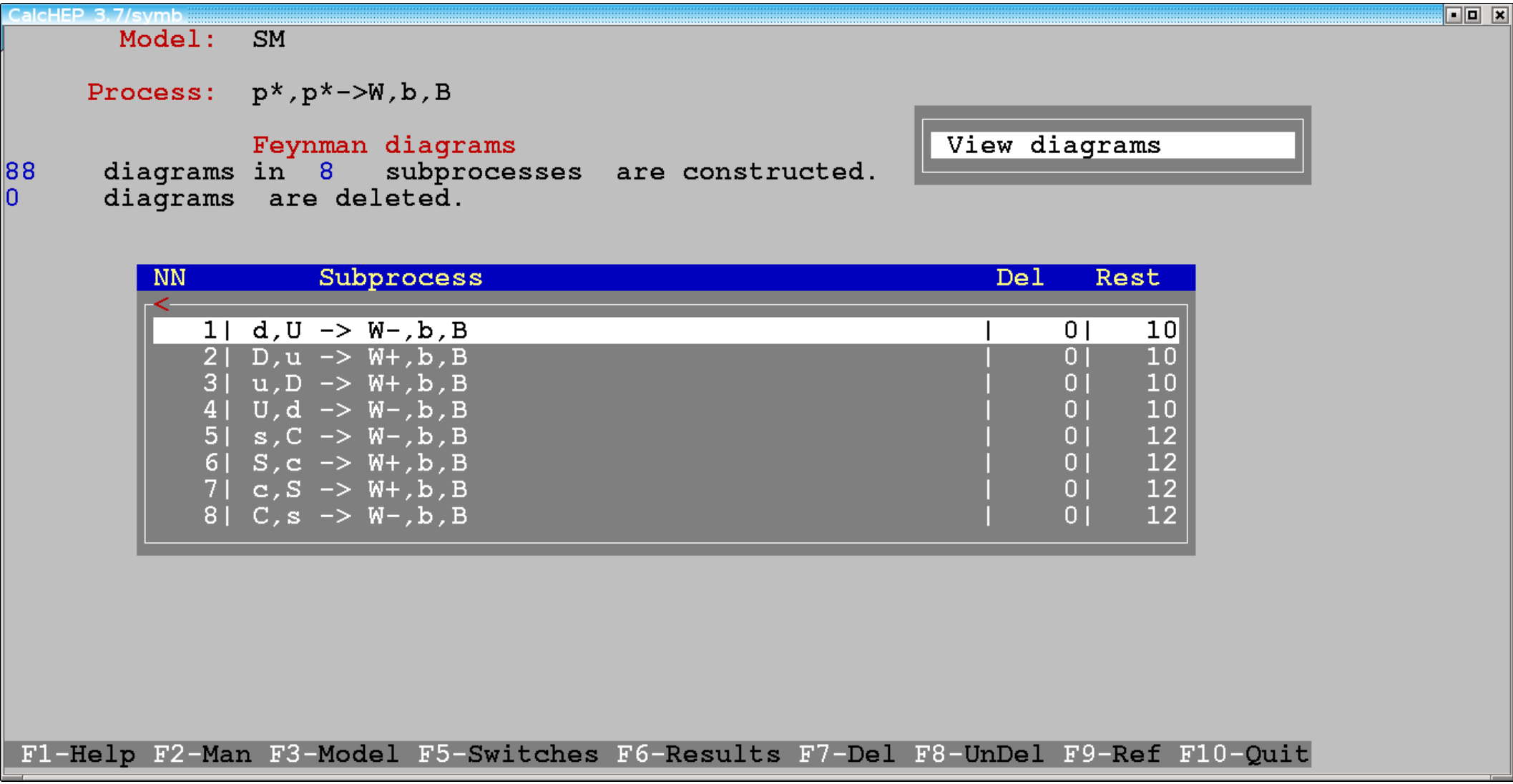
  'W+ > 2, A>1, Z>3'

ex#3: SM Higgs production cross section for e+e- → HZ process versus the collider energy for 0.5-1.0 TeV range

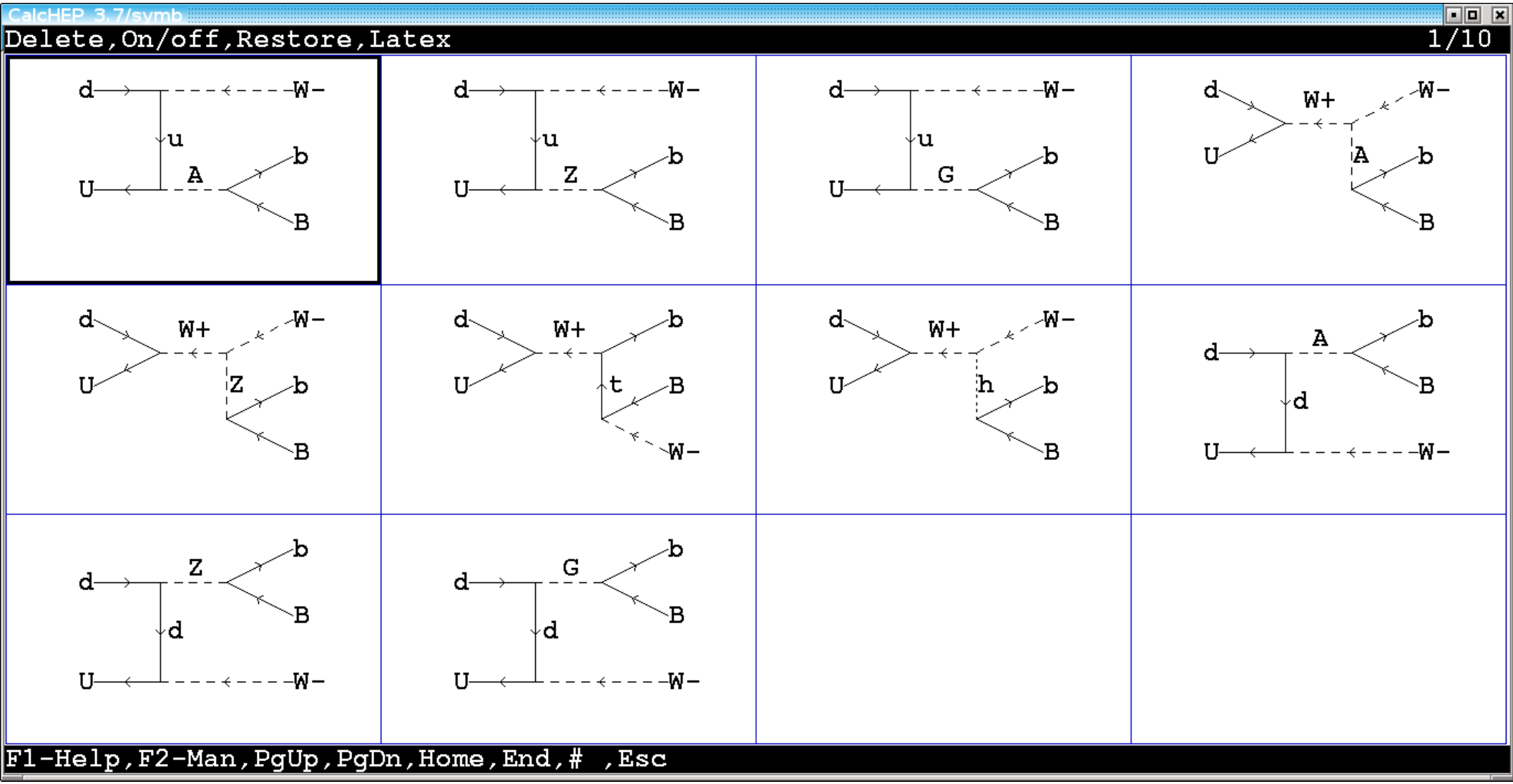
Symbolic session(1)



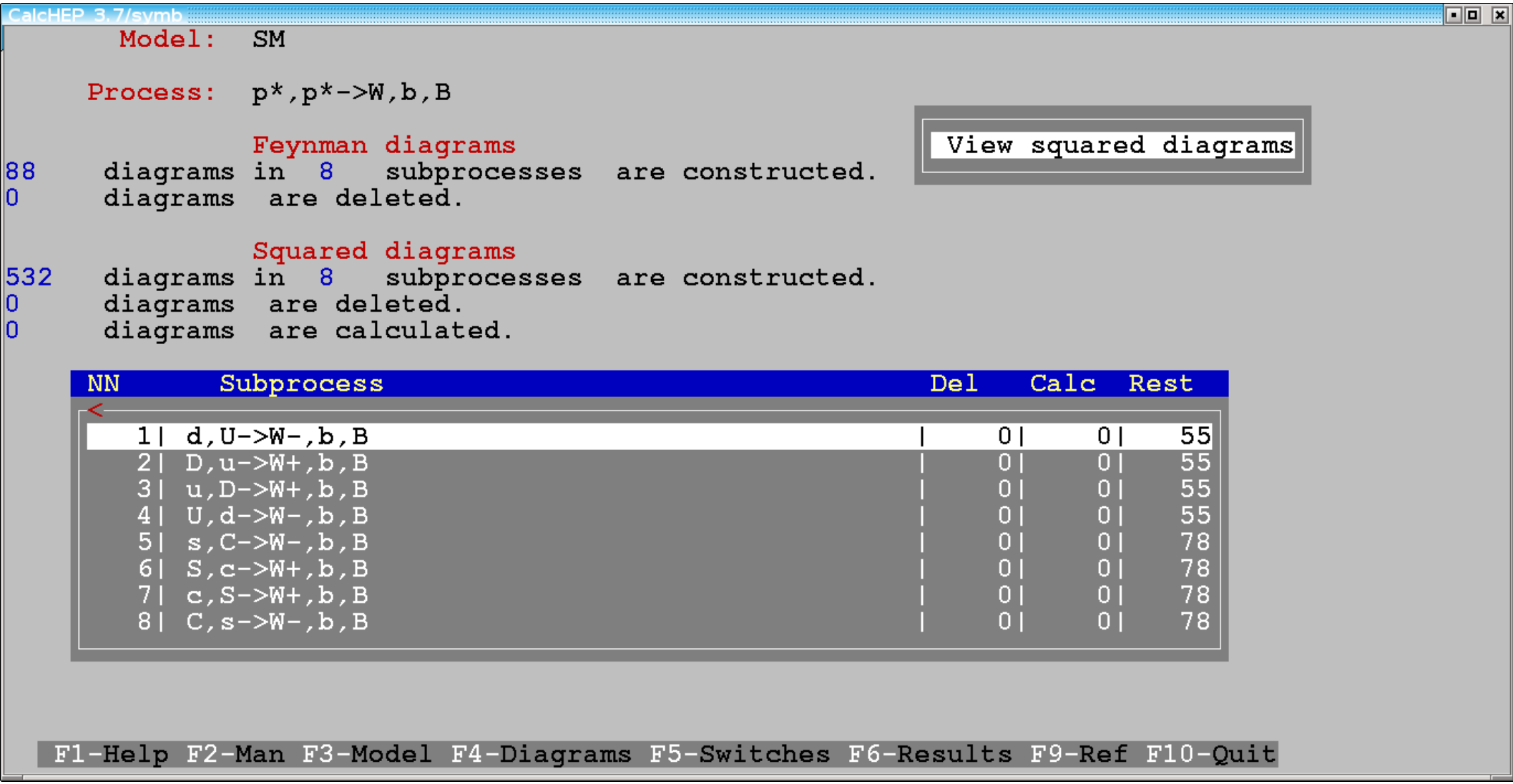
Symbolic session (2)



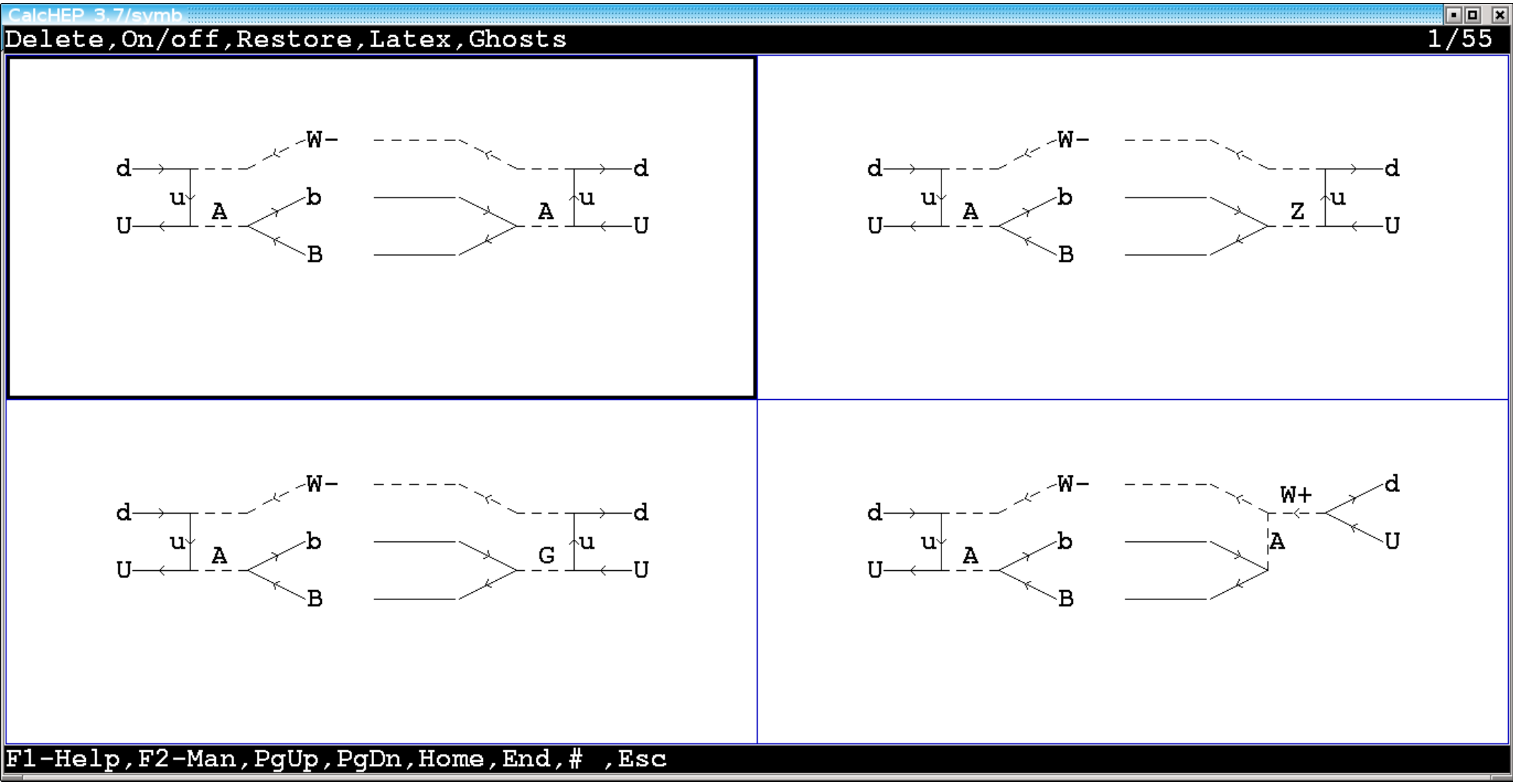
Symbolic session (3)



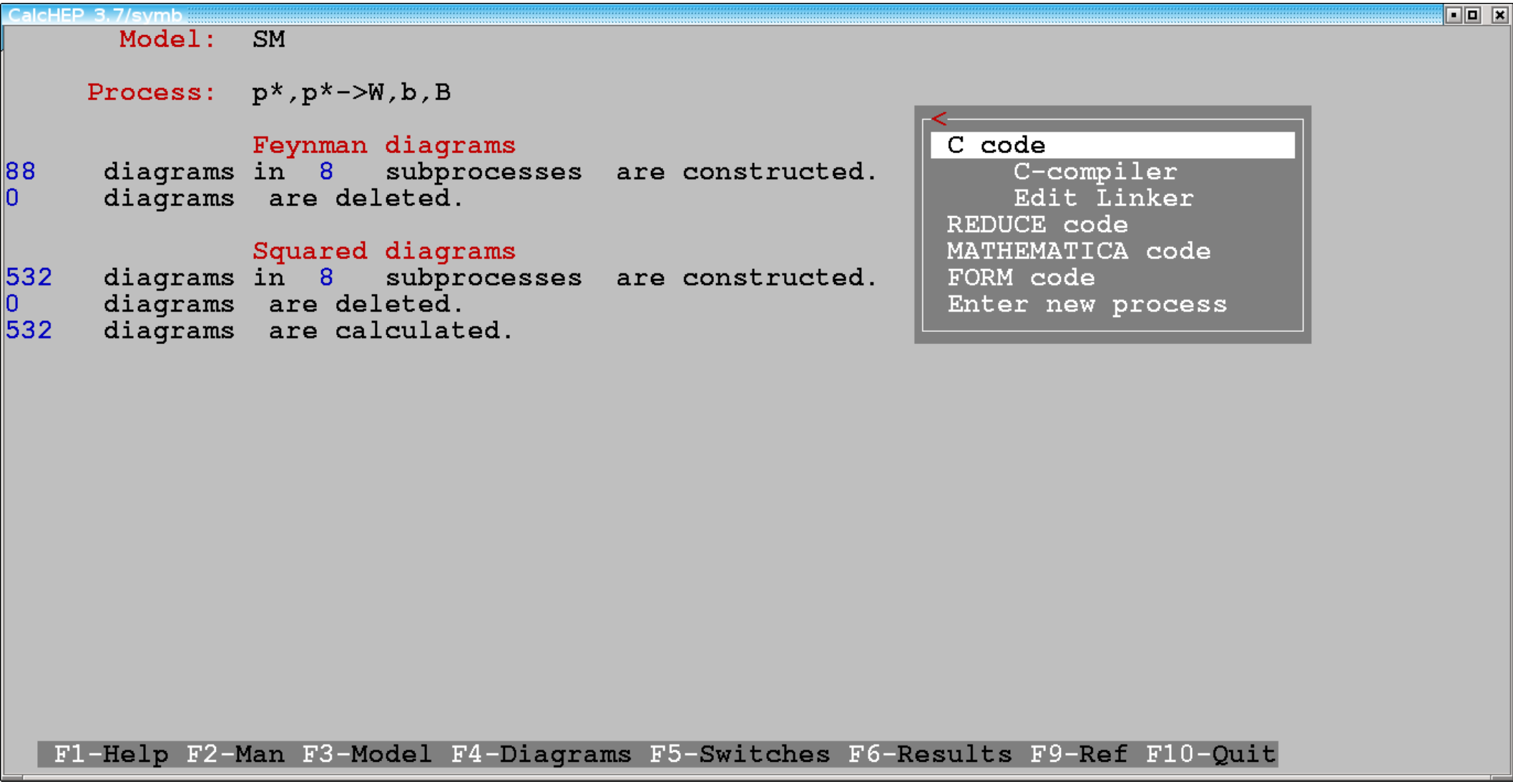
Symbolic session (4)



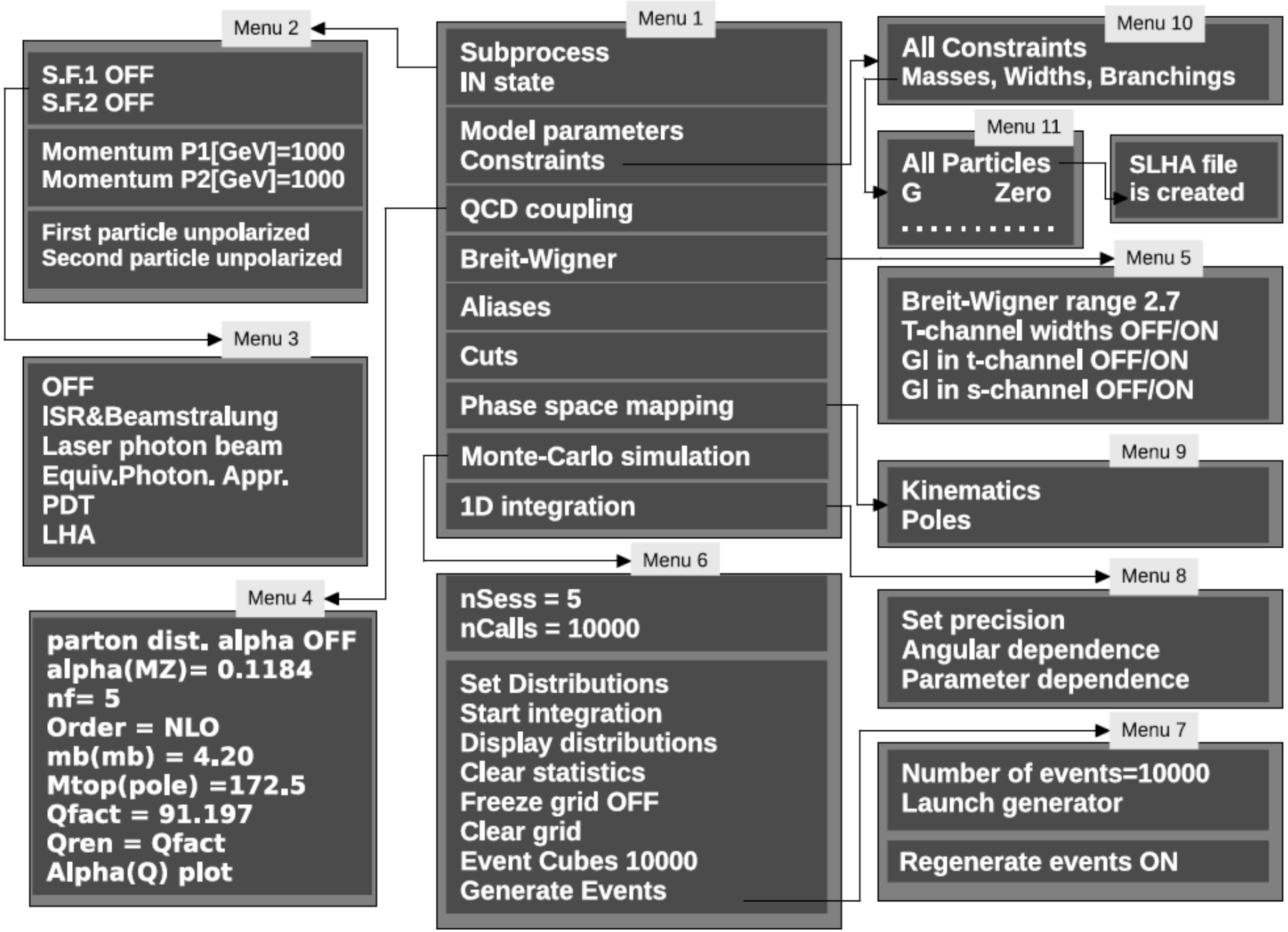
Symbolic session (5)



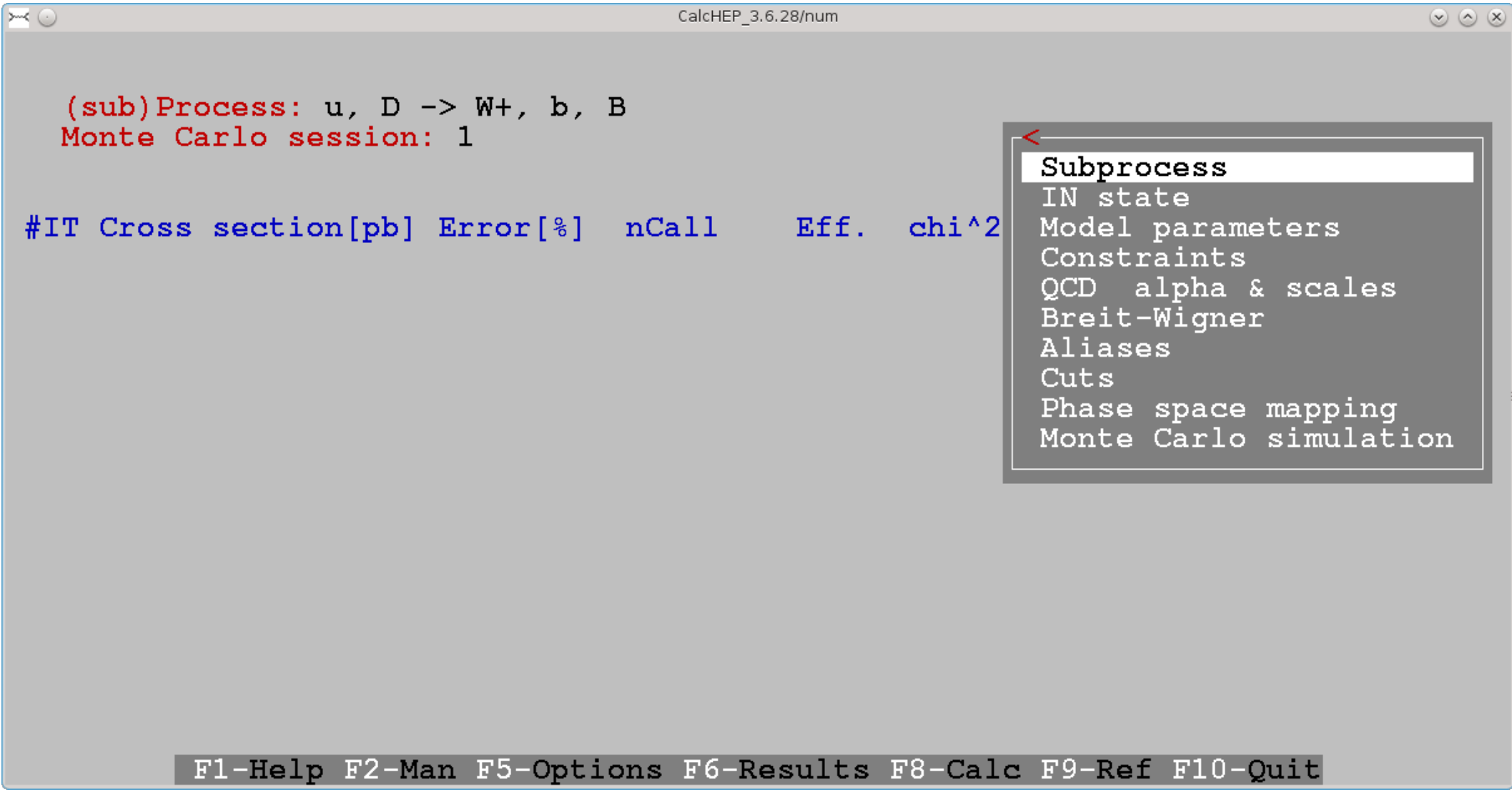
Symbolic session (6)



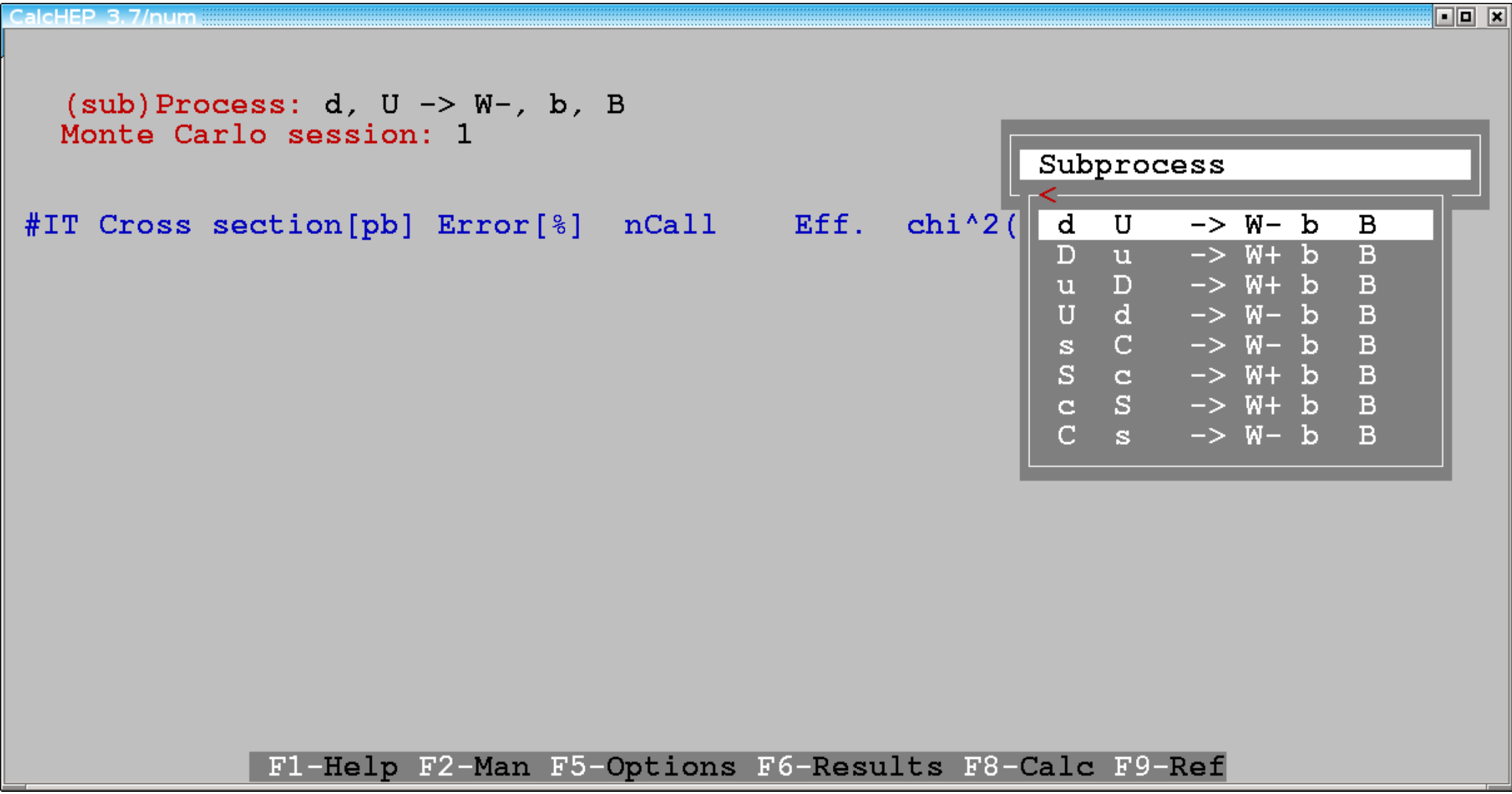
Structure of the numerical part



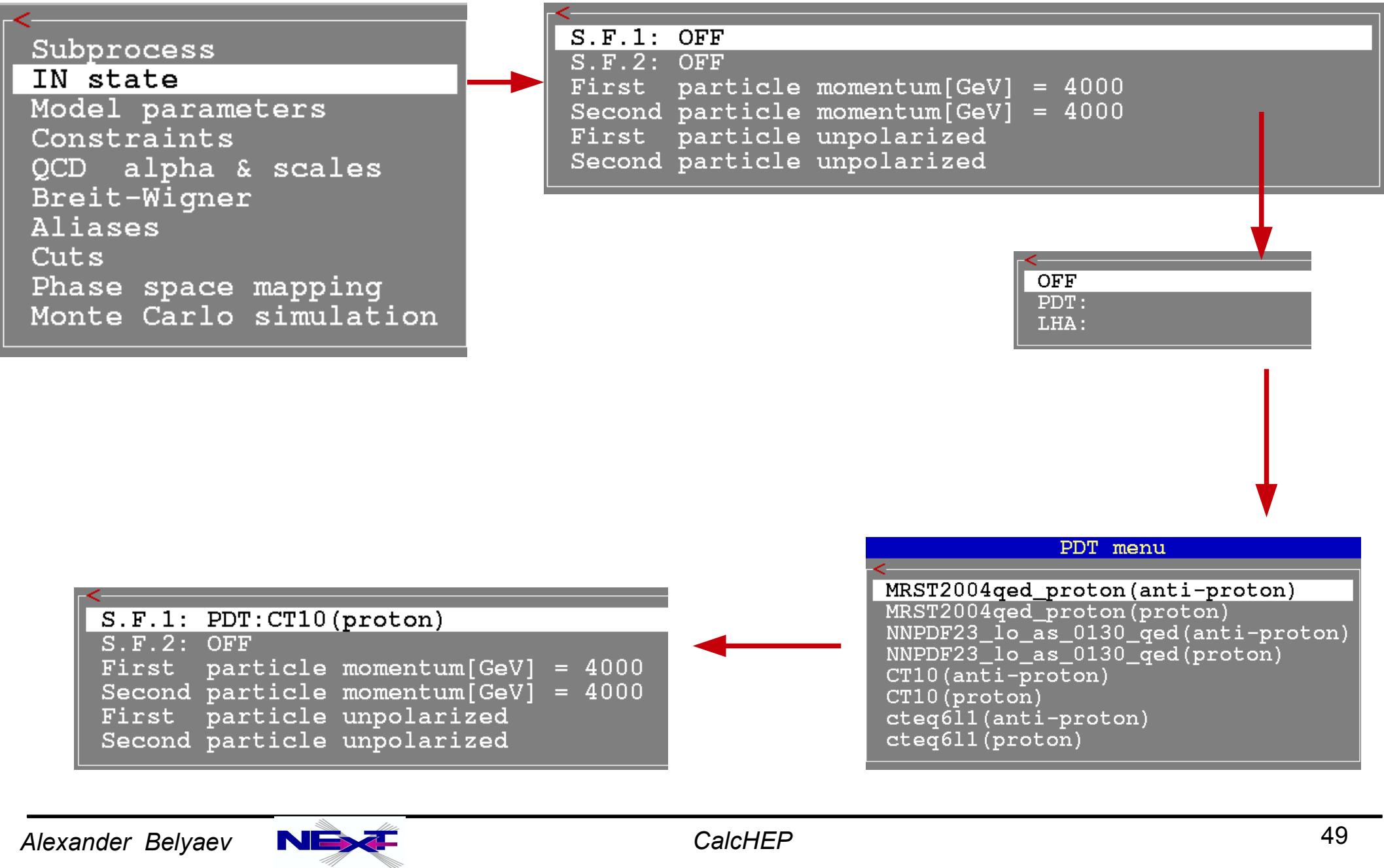
Numerical part(1)



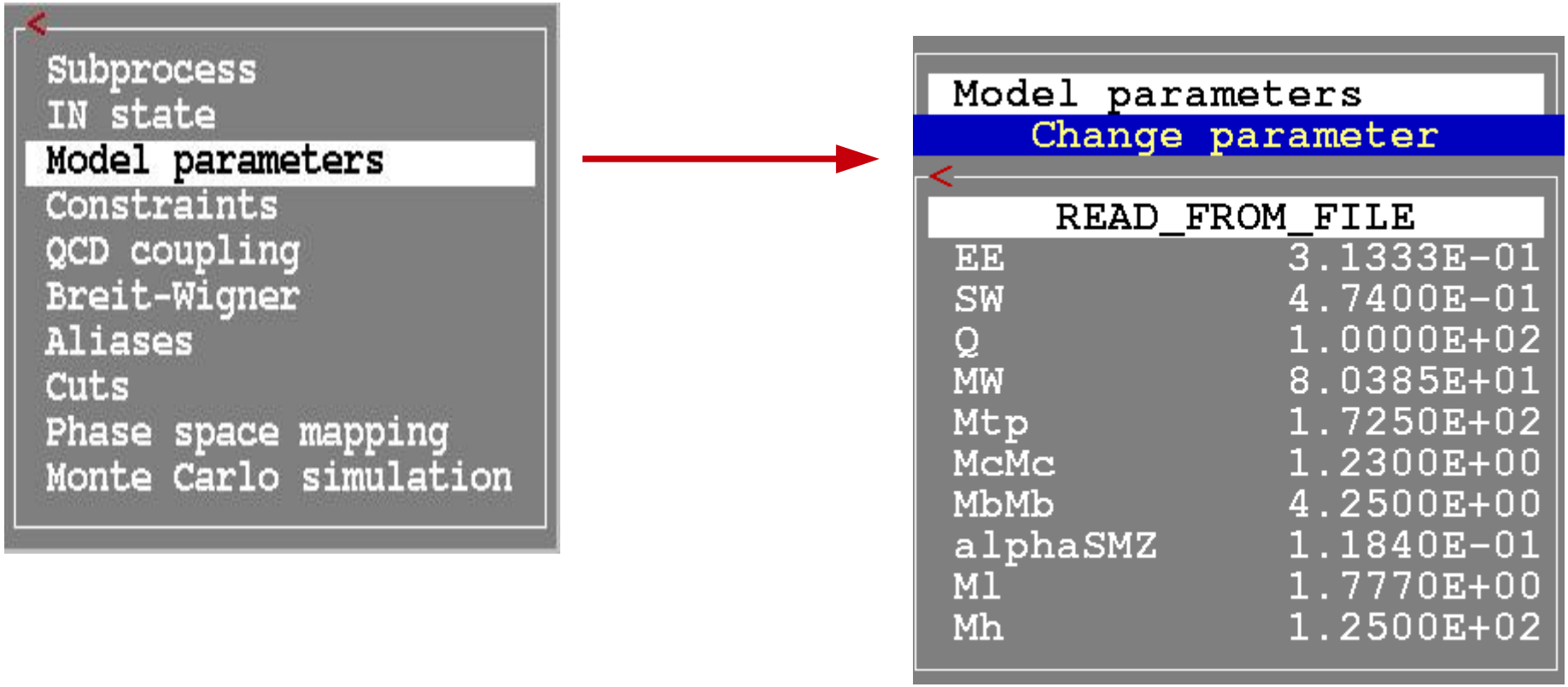
Numerical part(2)



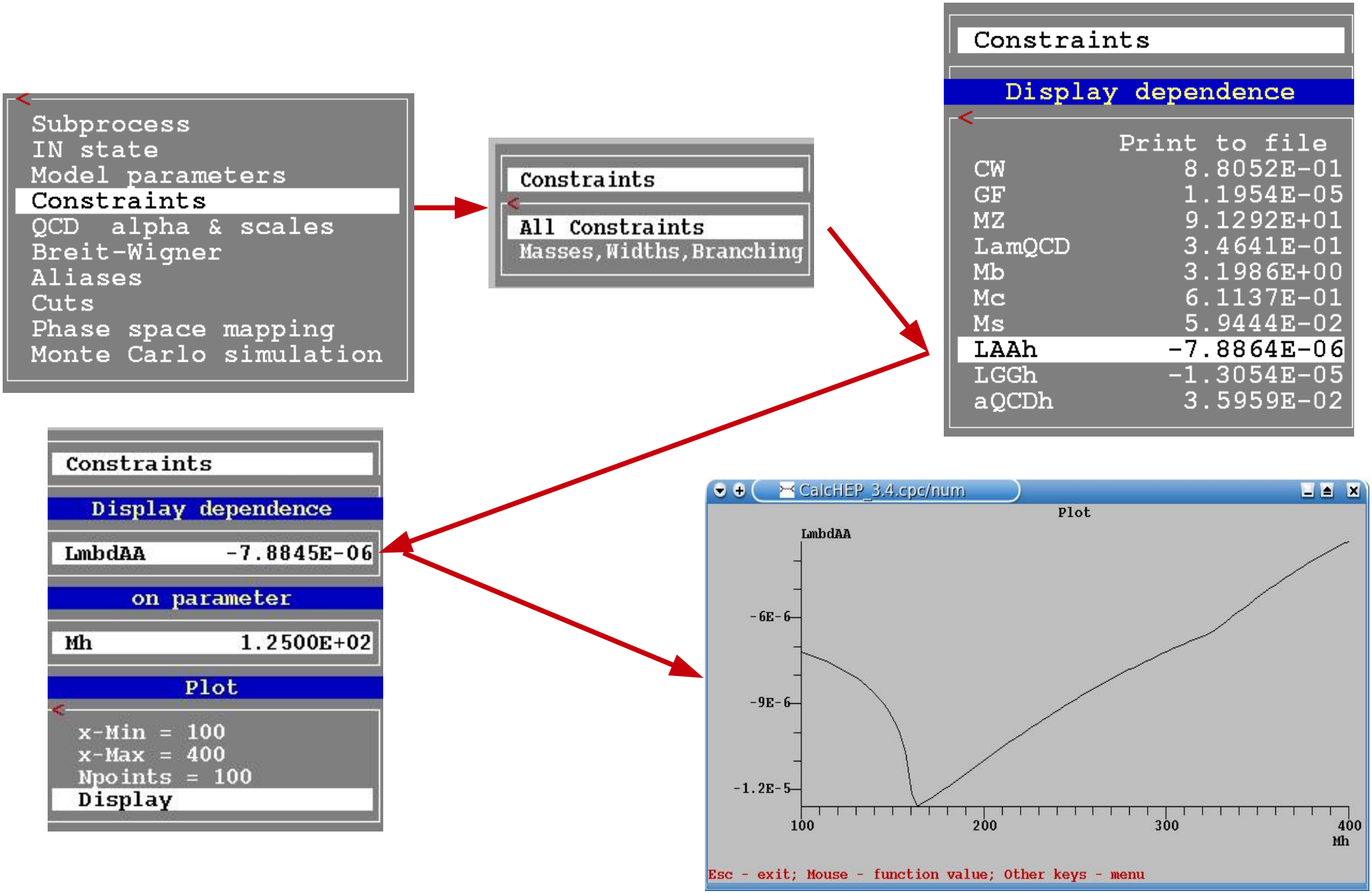
control of the initial states and parton density functions



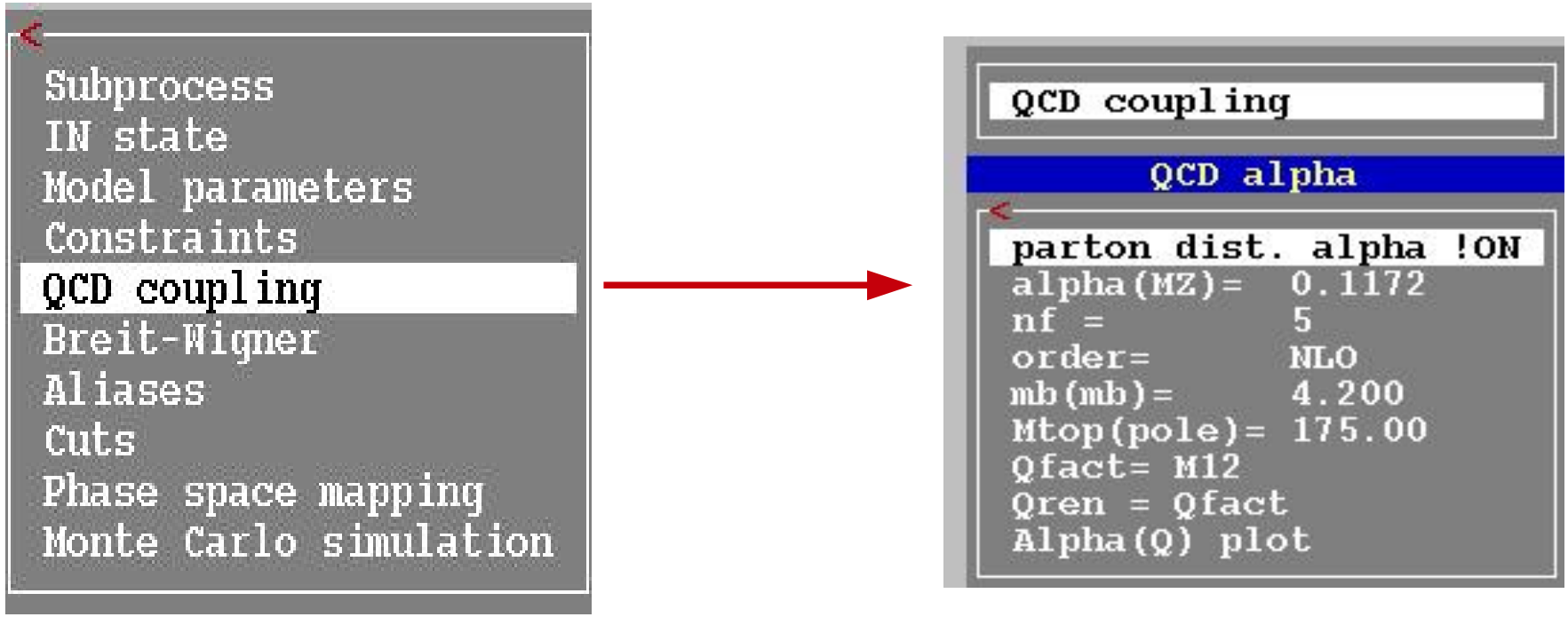
model parameters



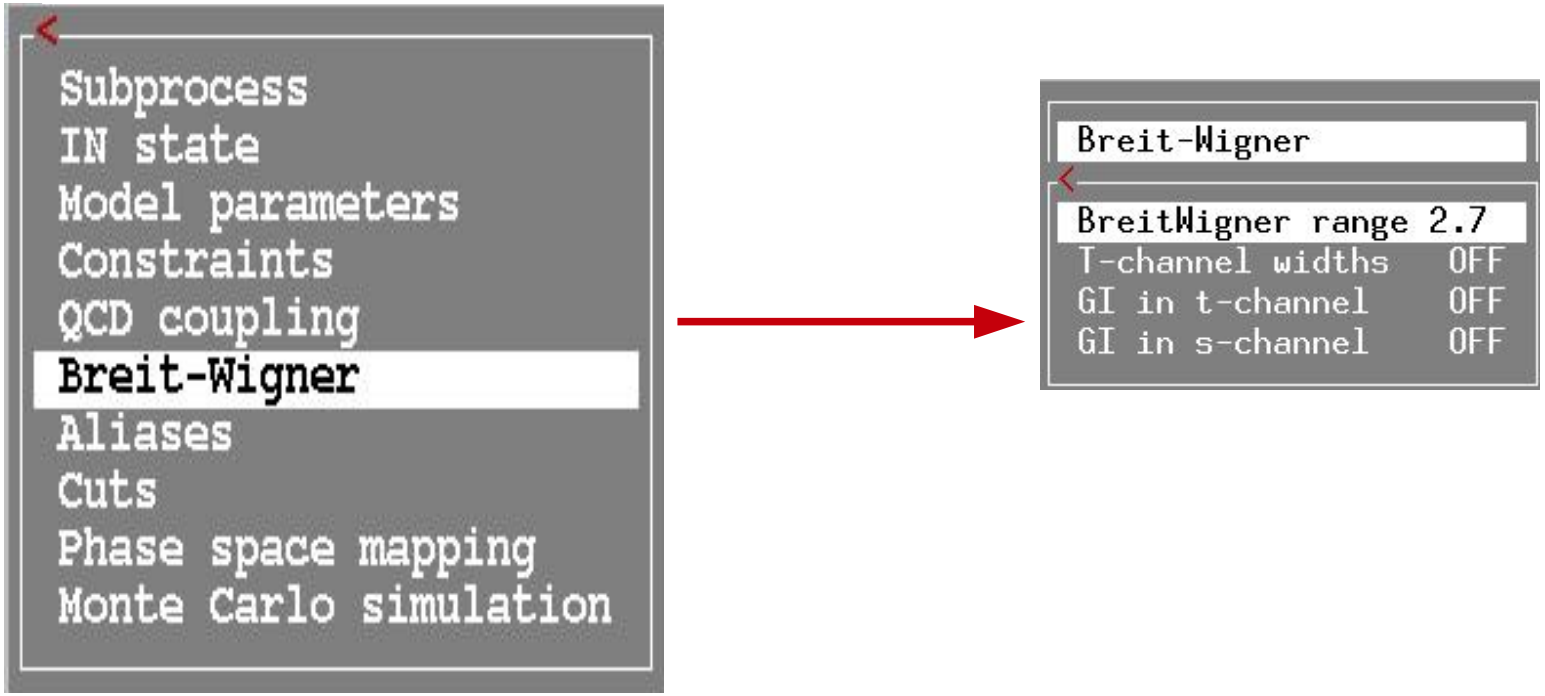
dependent parameters (SM+ggH model)



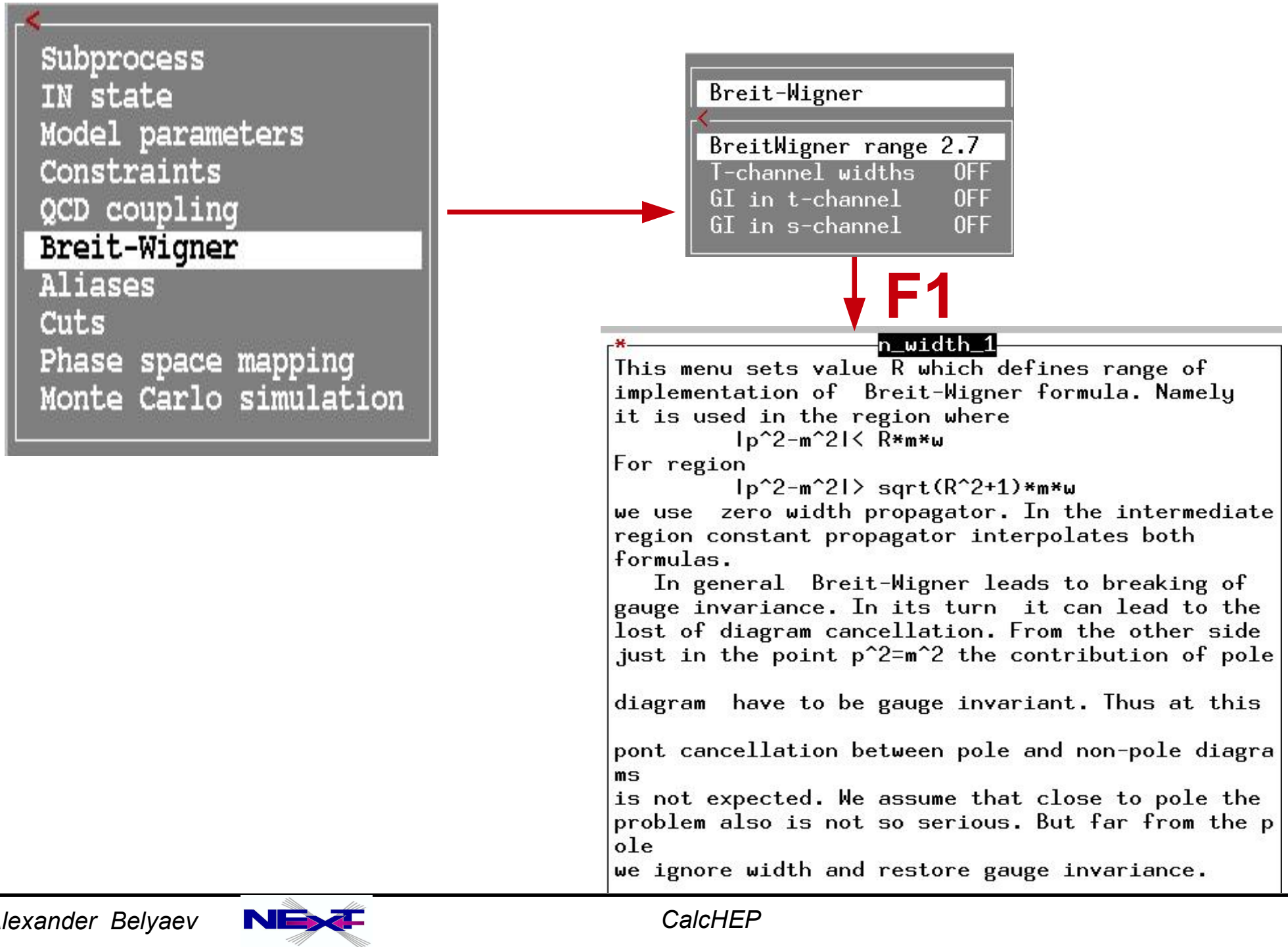
QCD coupling and the QCD scale



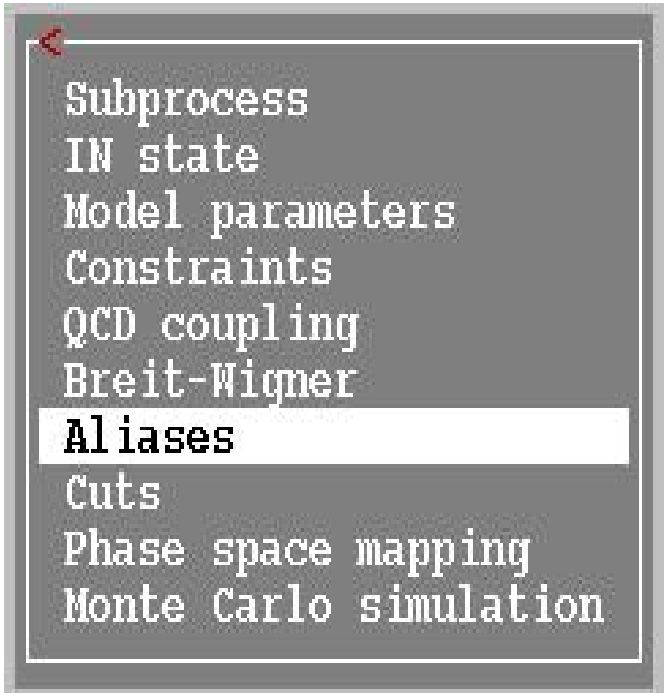
control of resonances

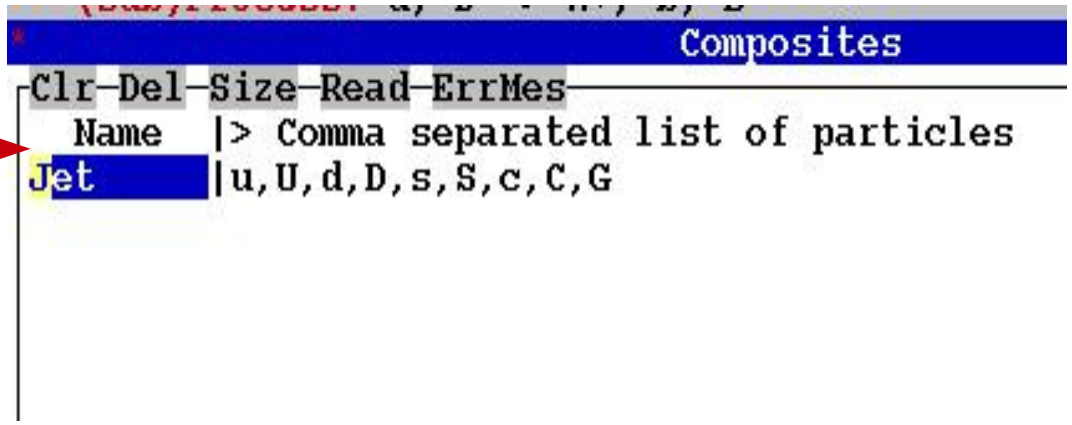


control of resonances

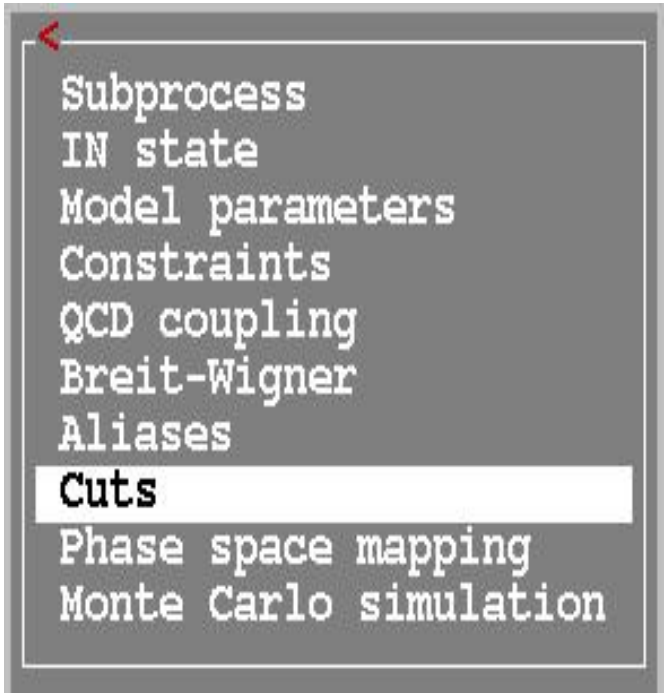


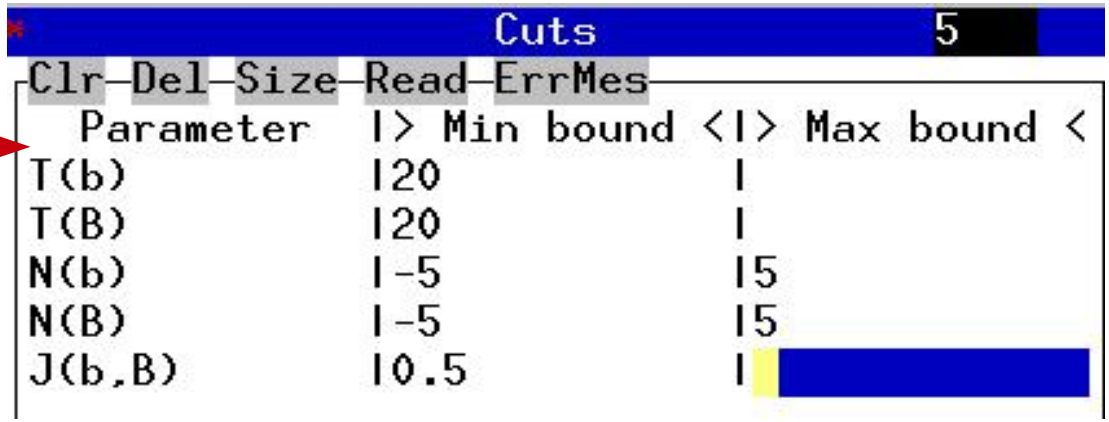
Aliases


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

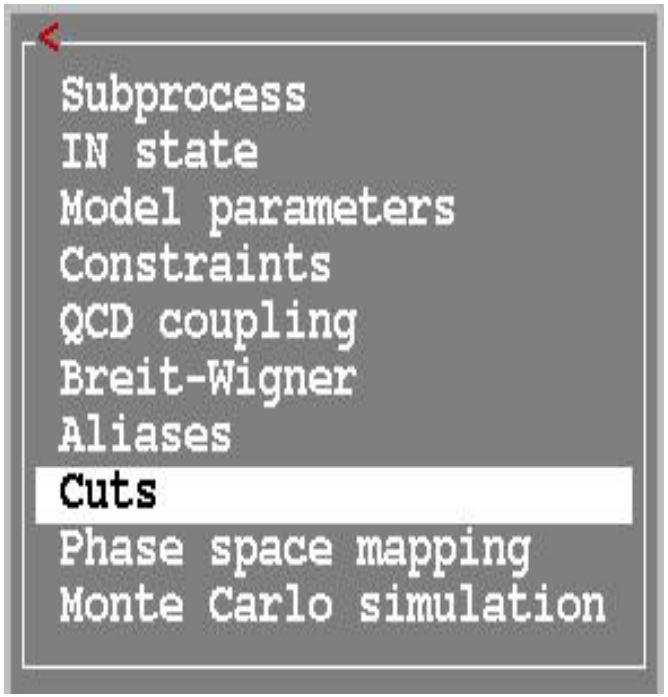

Composites
Clr Del Size Read ErrMes
Name |> Comma separated list of particles
Jet |u,U,d,D,s,S,c,C,G

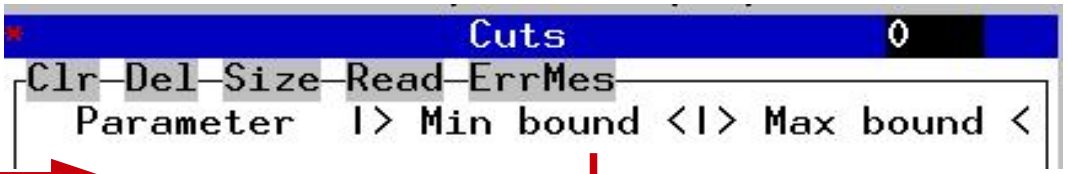
setting kinematical cuts





setting kinematical cuts





F1

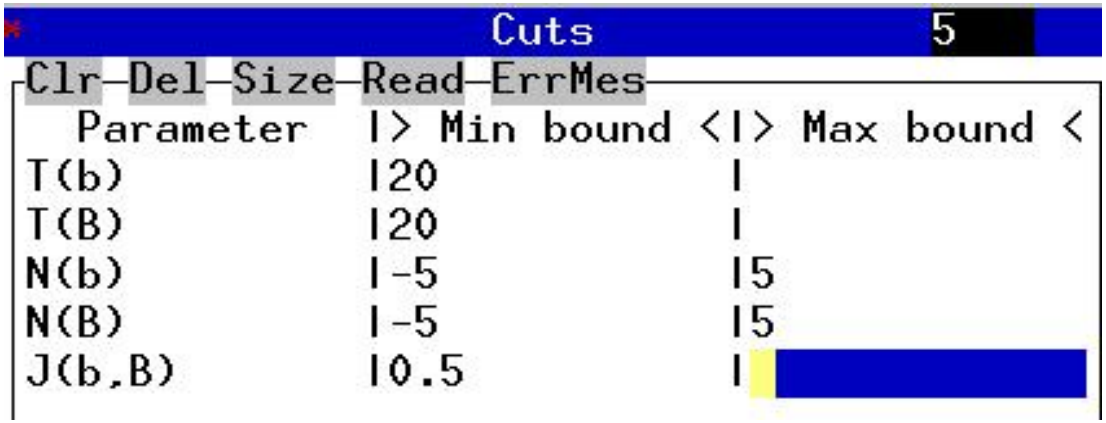
n_cut

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.

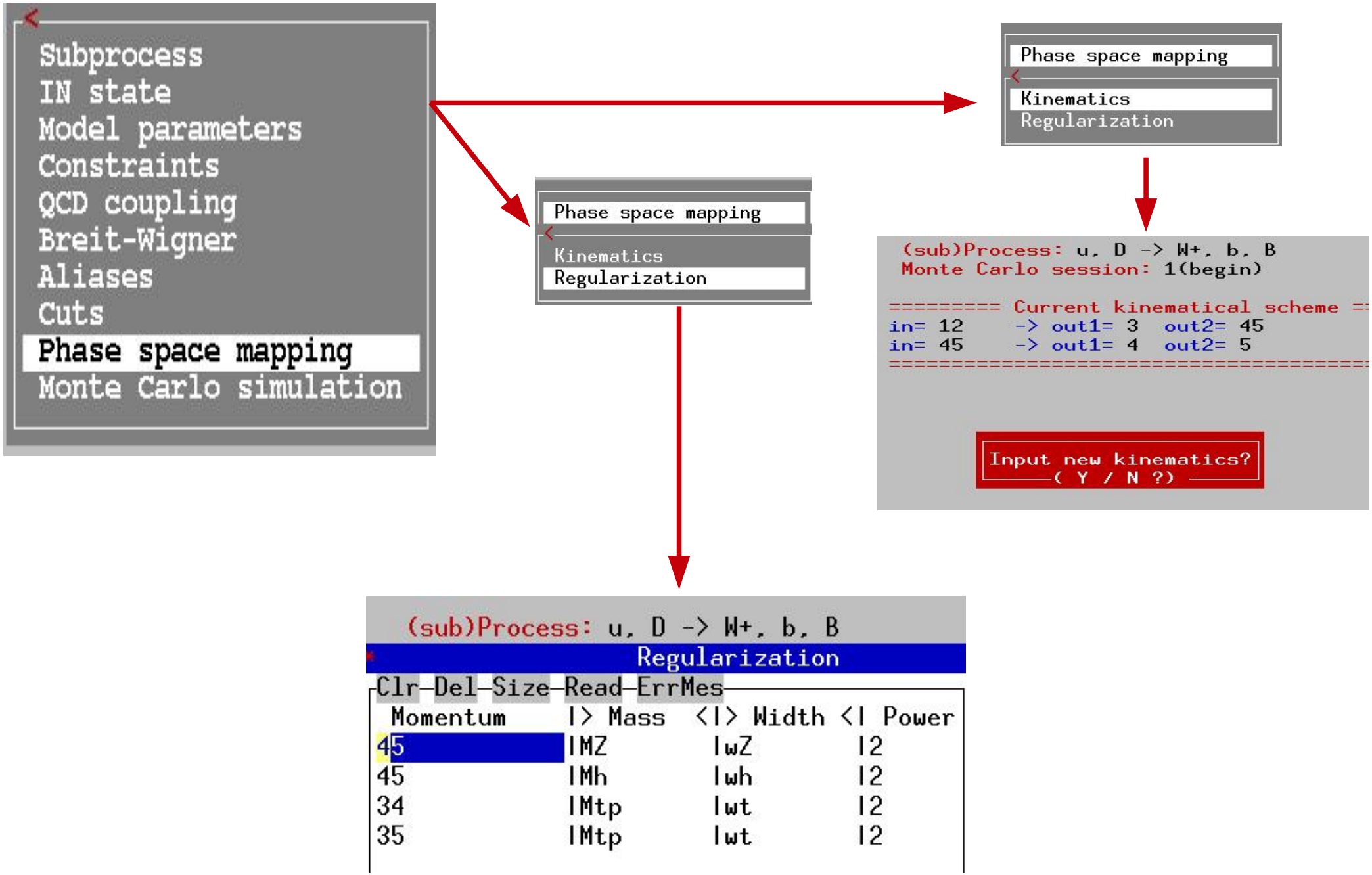
The phase space function is defined by its name which characterize type of cut and a particle list for which the cut is applied.
For example, "T(u)" means transverse momentum of 'u'-quark;
T(u,D) means summary transverse momentum of quark pair.
The following cut functions are available:

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-



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

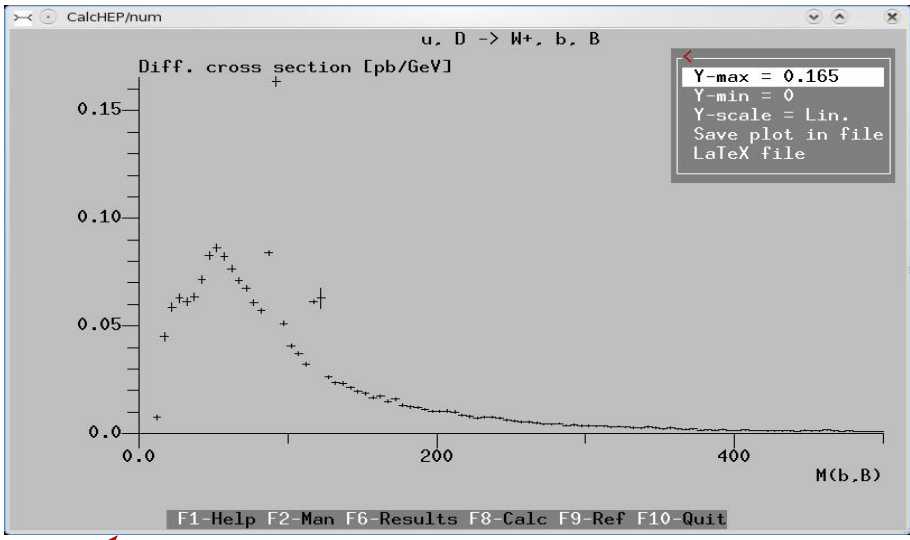
Clr	Del	Size	Read	ErrMes

Parameter_1	I> Min_1	<I> Max_1	<I Parameter_2	I> Min_2	<I> 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
```

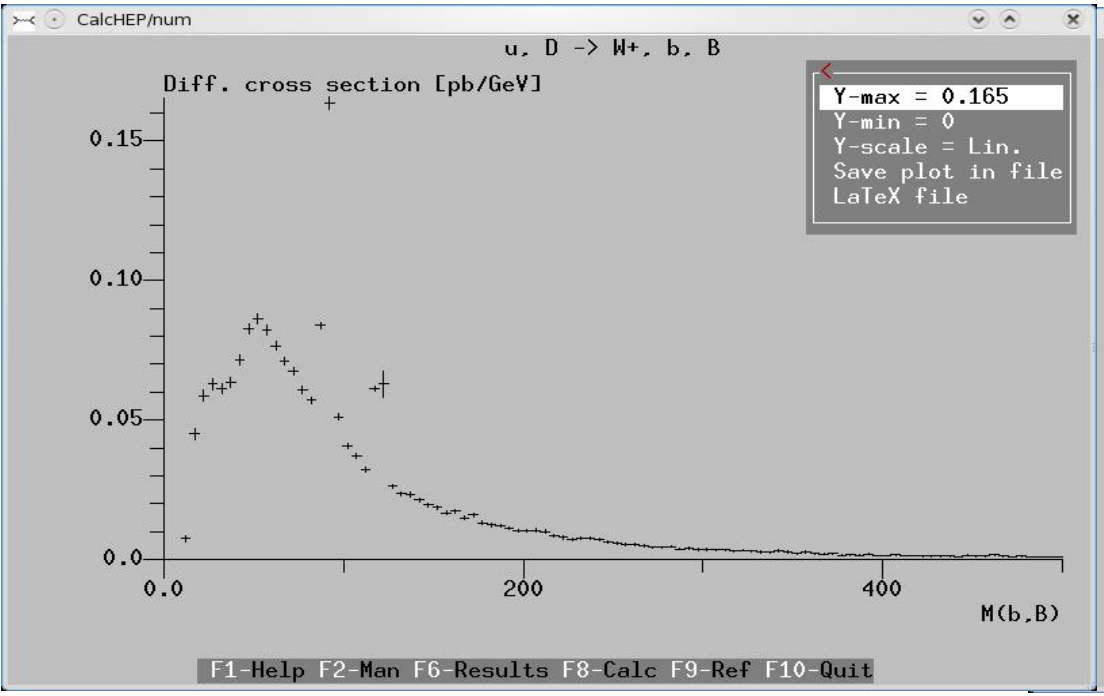
```
(sub)Process: u, D -> W+, b, B
Monte Carlo session: 2(continue)

#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      9.6945E+00        1.05E+00
 5      9.7032E+00        7.68E-01
< >     9.7095E+00        3.74E-01
```

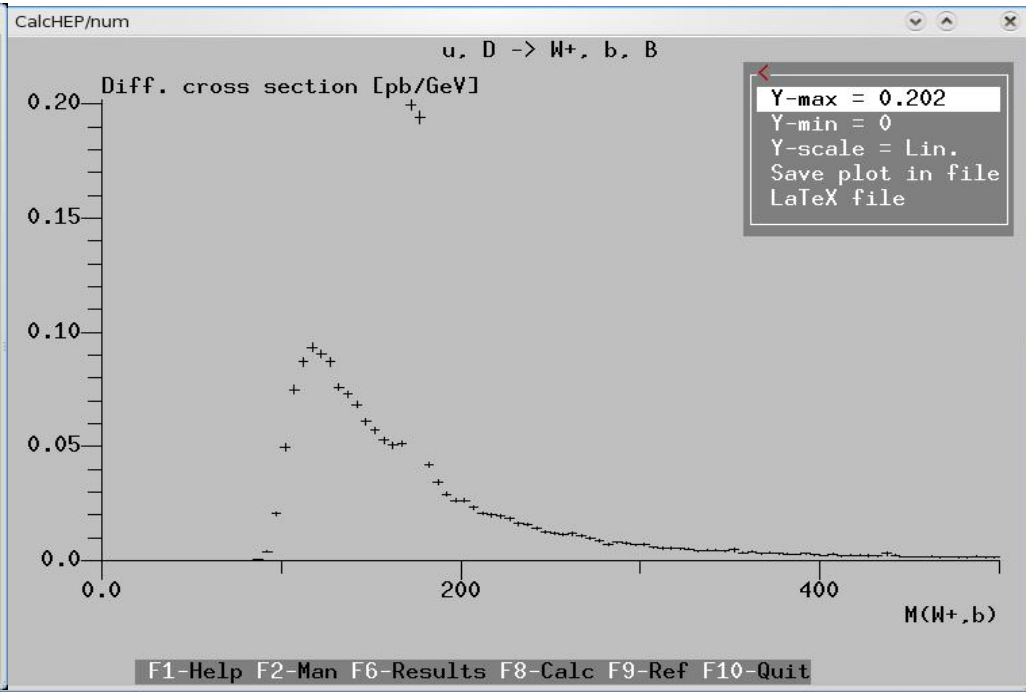
```
nSess  = 5
nCalls = 10000
Set  Distributions
*Start integration
Display Distributions
Clear statistic
Freeze grid          OFF
Clear  grid
Event Cubes 10000
Generate Events
```



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#4

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

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** ***lumi nmax***

  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

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

useful scripts for numerical session

see calchep_x.y/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

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

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: 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, e.g. calchep_3.7

```
cd  $CALCHEP/work
cp ../utile/batch_file .
./calchep_batch batch_file
```

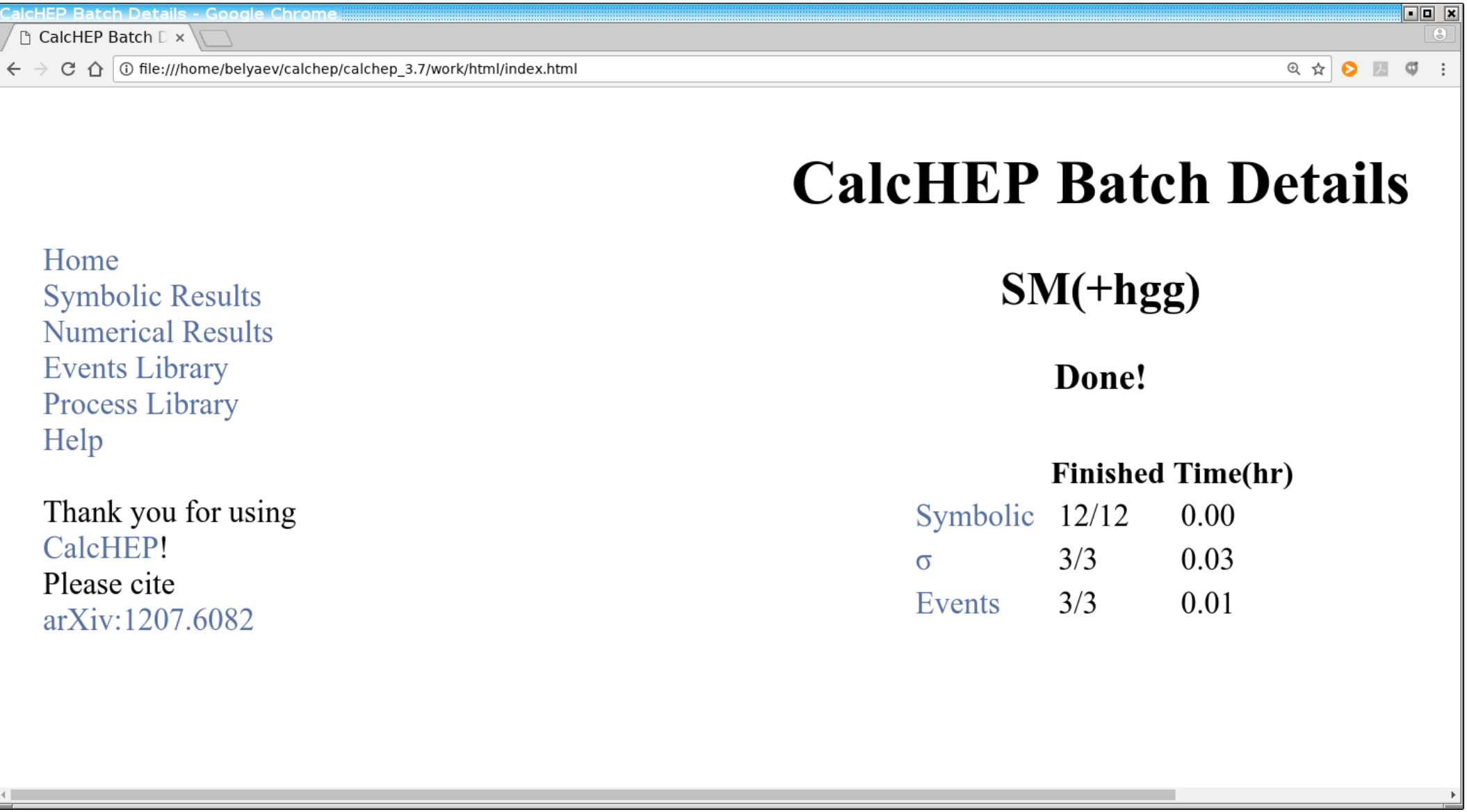
CALCHEP= /home/belyaev/calchep/calchep_3.7

calchep_batch version 1.41

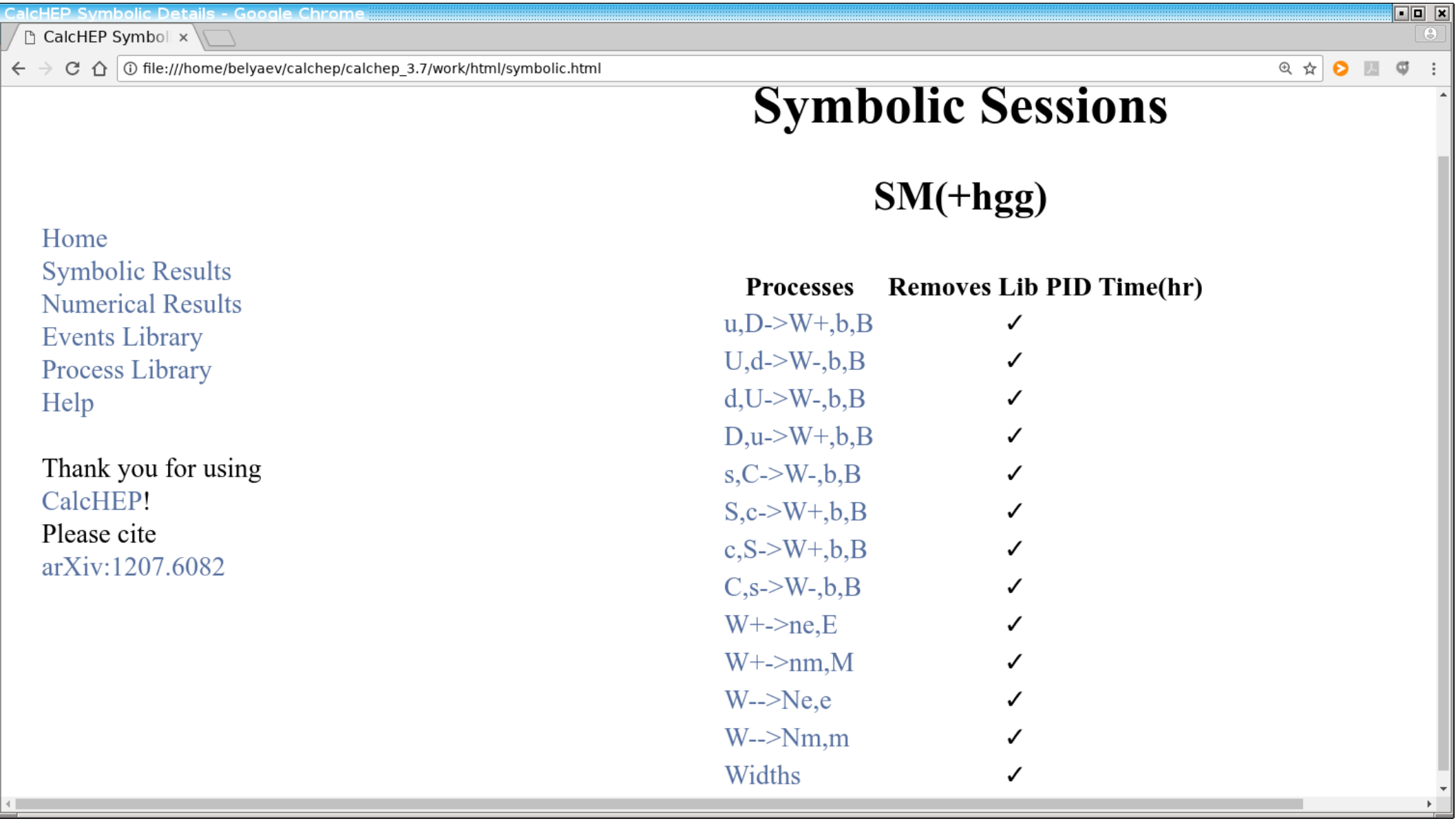
Processing batch:
Progress information can be found in the html directory.
Simply open the following link in your browser:
file:///.../calchep_3.7/work/html/index.html

You can also view textual progress reports in
.../calchep_3.7/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

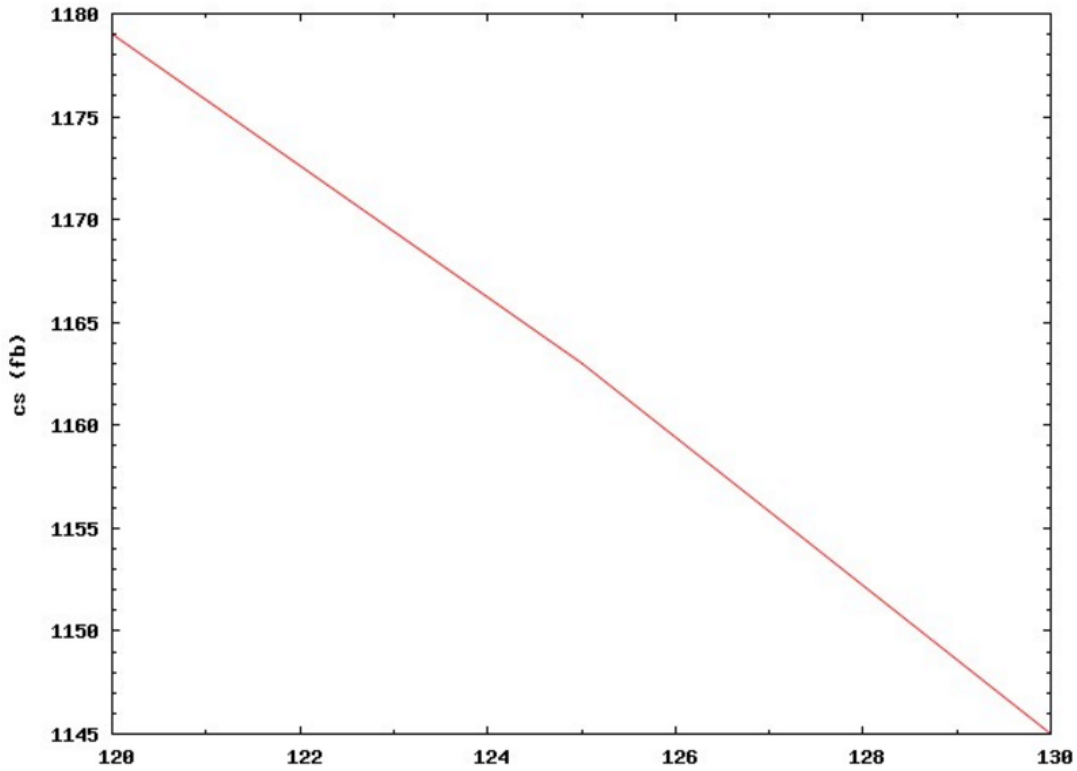
SM(+hgg)

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)

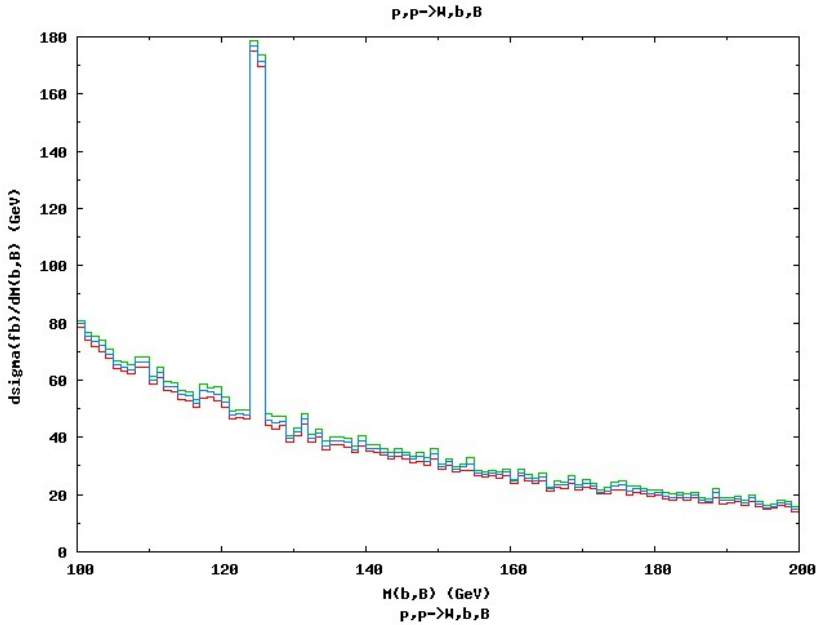
Done!

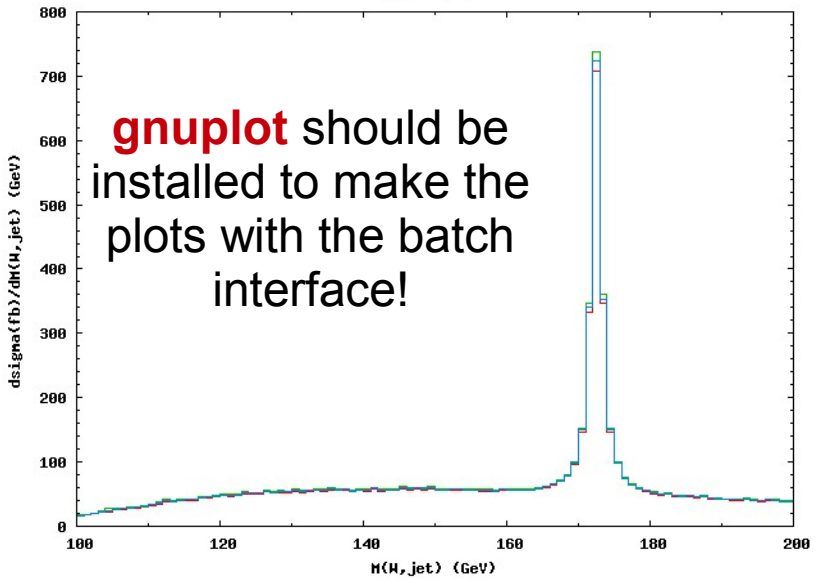
Processes	σ (fb)	Δσ (%)	PID	Time (hr)	N events	Details
u,D->W+,b,B	1552.8	0.8	28872	0.00	383/382	prt_1 session.dat
U,d->W-,b,B	829.4	0.52	28878	0.00	220/219	prt_1 session.dat
d,U->W-,b,B	837.46	1.1	28885	0.00	221/220	prt_1 session.dat
D,u->W+,b,B	1558.3	0.51	29100	0.00	384/383	prt_1 session.dat
s,C->W-,b,B	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#6: *using calchep_batch evaluate complete cross section for pp-> Wbb process with the same cuts as for ex#4*

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

see https://answers.launchpad.net/calchep for many "tricky" questions/answers

Model implementation in CalcHEP

using LanHEP (*Andrei Semenov*)

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

- **To install**
  wget https://theory.sinp.msu.ru/~semenov/lhep332.tgz
  tar -zxvf lhep*332*.tgz
  cd lanhep*332*
  make

- **To Run**
  cd mdl
  ../lhep -ca stand.mdl

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

- **Also you can do**
  ../lhep -ufo stand.mdl
  ../lhep -tex stand.mdl

  to produce model in UFO format and get Feynman rules in the LaTeX format respectively

QCD as an example

- **Gauge interactions** $L_{YM} = -\frac{1}{4} F^{a\mu\nu} F^a_{\mu\nu},$

where $F^a_{\mu\nu} = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu - g_s f^{abc} G^b_\mu G^c_\nu,\ G^a_\mu(x)$

- **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

QCD Feynman rules generated by LanHEP in LaTeX format

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

$H\gamma\gamma, Hgg$ Effective vertices in CalcHEP

- **Model sources in CalcHEP**
  cd $CALCHEP/model_src
  ls
  sm.inc, sm.lhep, idm.lhep

  $LANHEP/lhep -ca sm
  $LANHEP/lhep -ca idm
  will produce SM and IDM models for CalcHEP

- **The sm.inc contains the $H\gamma\gamma$ implementation**

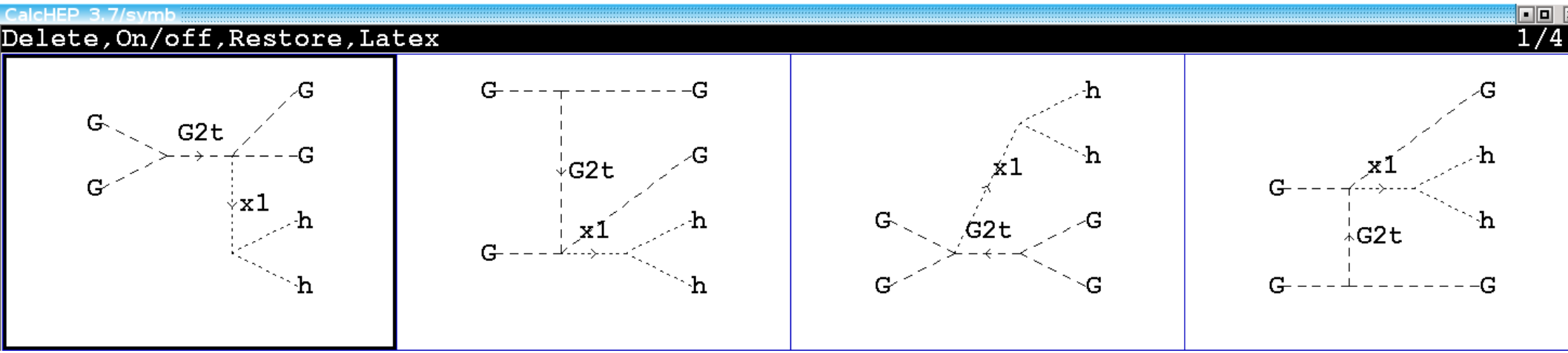
```
let shd = {  i*'W+.f',  (vev(vevh)+h-i*'Z.f')/Sqrt2 }.
external_func(lAAhiggs,2).
parameter LAAh=-cabs(lAAhiggs(Mh,str(h))).
lterm  LAAh*(shd*anti(shd)-vevh**2/2)/vevh*F**2  where
         F=deriv^mu*A^nu-deriv^nu*A^mu.
```

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

Introducing auxiliary particles in CalcHEP

- $H^2 F^{\mu\nu} F_{\mu\nu}$ ***effective Lagrangian leads to 6-point*** **Hhgggg** ***vertex, CalcHEP 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.
```



High Energy Physics Model Database

https://hepmdb.soton.ac.uk/

(supported by IPPP in 2012-13)

- collects HEP models and model sources in all formats
- allows you to upload your models (public or private mode)
- allows you to evaluate processes (CalcHEP, Madgraph, Whizard) and perform event generation for all models at HEPMDB using HPC cluster
  using simple web interface
- you can preform calculations using web interface avoiding problems related to installing the actual software, which can sometimes be quite cumbersome

[The rest will be in CalcHEP tutorial]

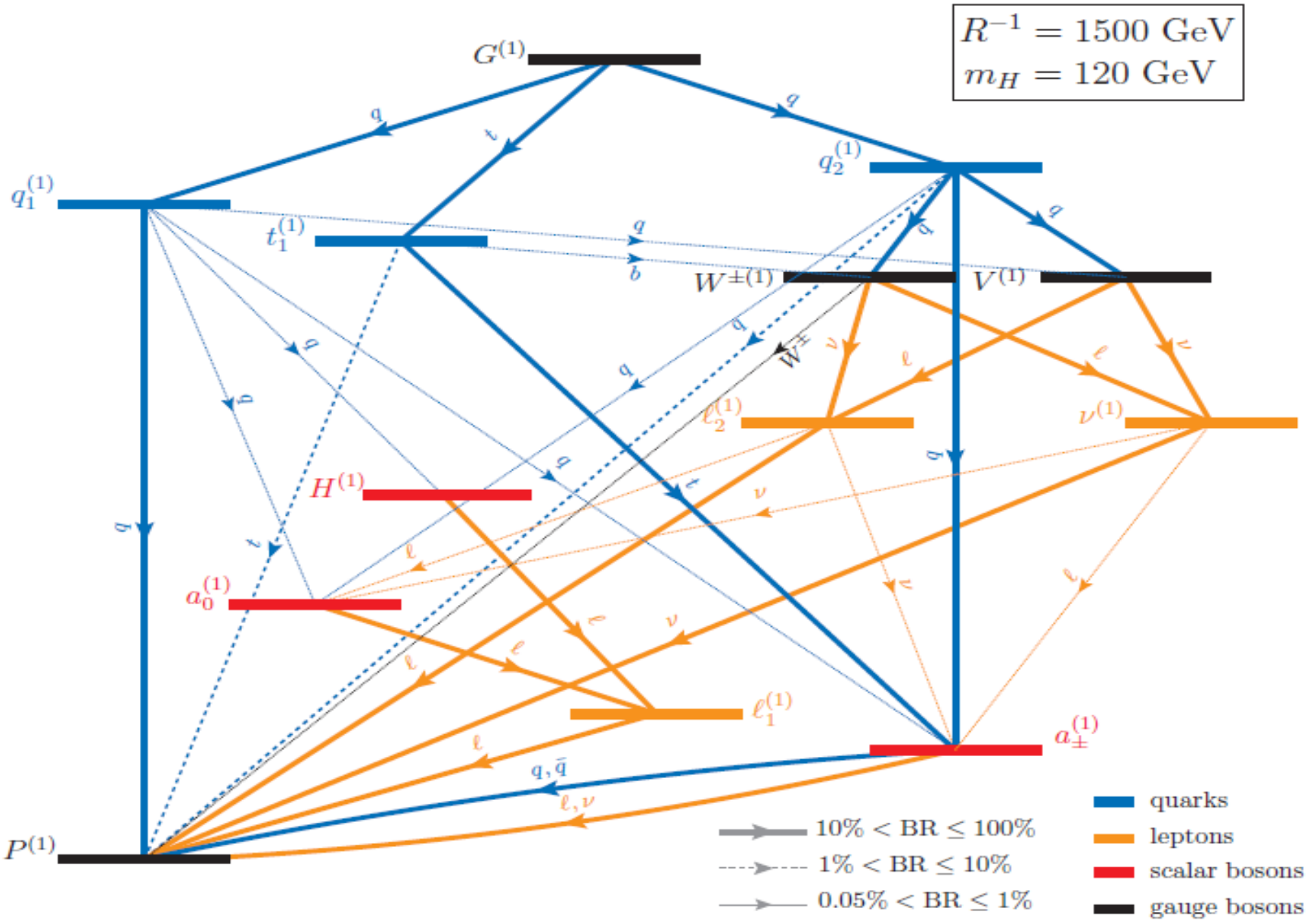
Example of models created for CalcHEP

- **SM + extensions**
  - SM
  - B-L symmetric Z' with heavy Majorana neutrinos
  - SM + Z'
  - general 2 Higgs doublet model
  - 4th generation
  - Excited fermions
  - Model with contact interactions
  - Standard Model + anomalous gauge boson couplings
  - Model of strongly int EW sector
    (5 & 6 dim operators involving Sigma field)
- **SUSY**
  - constraint MSSM
  - general MSSM, with 124 free parameters
  - NMSSM
  - RPVMSSM
  - left-right symmetric MSSM
  - MSSM with CP violation
  - E6MSSM
- **Extra dimensions**
  - 5D UED with 2KK layers
  - 6D UED with 2KK layers
  - ADD = ADD
  - RS = Randall Sundrum
- **Leptoquarks**
  - Complete LQ model
    SU(3)xSU(1)xU(1) vector&scalar
- **Technicolor & Higgsless**
  - Minimal walking technicolor
  - TC with DM
  - 3-site model
  - Hidden Local symmetry model
  - 4SM = general 4-site model
- **Little Higgs**
  - Littlest higgs model with T-parity
  - LHT + T-parity violation

One of the examples

Universal Extra Dimensions

hep-ph/1212.4858 ***In collaboration with M.Brown, J.M. Moreno, C. Papineau***



Universal Extra Dimensions

- **Set up of the production and decay processes with the calchep_batch**

```
Process: p,p->y2,y2
Process: p,p->y3,y3
Process: p,p->y2,y3

Decay: y1->2*x
Decay: y2->2*x
Decay: y3->2*x
Decay: y4->2*x
Decay: y5->2*x
Decay: y6->2*x
Decay: y7->2*x
Decay: y8->2*x

Composite: p=u,U,d,D,s,S,c,C,b,B,G
Composite: y1=~G_1
Composite: y2=~d1_1,~u1_1,~s1_1,~c1_1,~b1_1,~t1_1,~d2_1,~u2_1,~s2_1,~c2_1,~b2_1,~t2_1
Composite: y3=~D1_1,~U1_1,~S1_1,~C1_1,~B1_1,~T1_1,~D2_1,~U2_1,~S2_1,~C2_1,~B2_1,~T2_1
Composite: y4=Z,W+,W-,t,T,H
Composite: y5=~P_1,~V_1,~W+_1,~W-_1
Composite: y6=~e1_1,~e2_1,~n1_1,~mu1_1,~mu2_1,~n2_1,~tau1_1,~tau2_1,~n3_1
Composite: y7=~E1_1,~E2_1,~N1_1,~Mu1_1,~Mu2_1,~N2_1,~Tau1_1,~Tau2_1,~N3_1
Composite: y8=~H_1,~a0_1,~a+_1,~a-_1
```

- **Scan in 2D space with the calchep_batch**

```
################################################
#  Run Info                                    #
#  Masses and Energies are in GeV              #
#  More than one run can be specified at       #
#     the same time.                           #
################################################
Run parameter: invR
Run begin:     600
Run step size: 200
Run n steps:   4
Run parameter: nL
Run begin:     10
Run step size: 10
Run n steps:   4
```

Results from calchep_batch at HEPMDB

Home
Symbolic Results
Numerical Results
Events Library
Process Library
Help

Thank you for using CalcHEP!
Please cite arXiv:0000.0000

CalcHEP Batch Details

MUED-Chloe-2KK

Done!

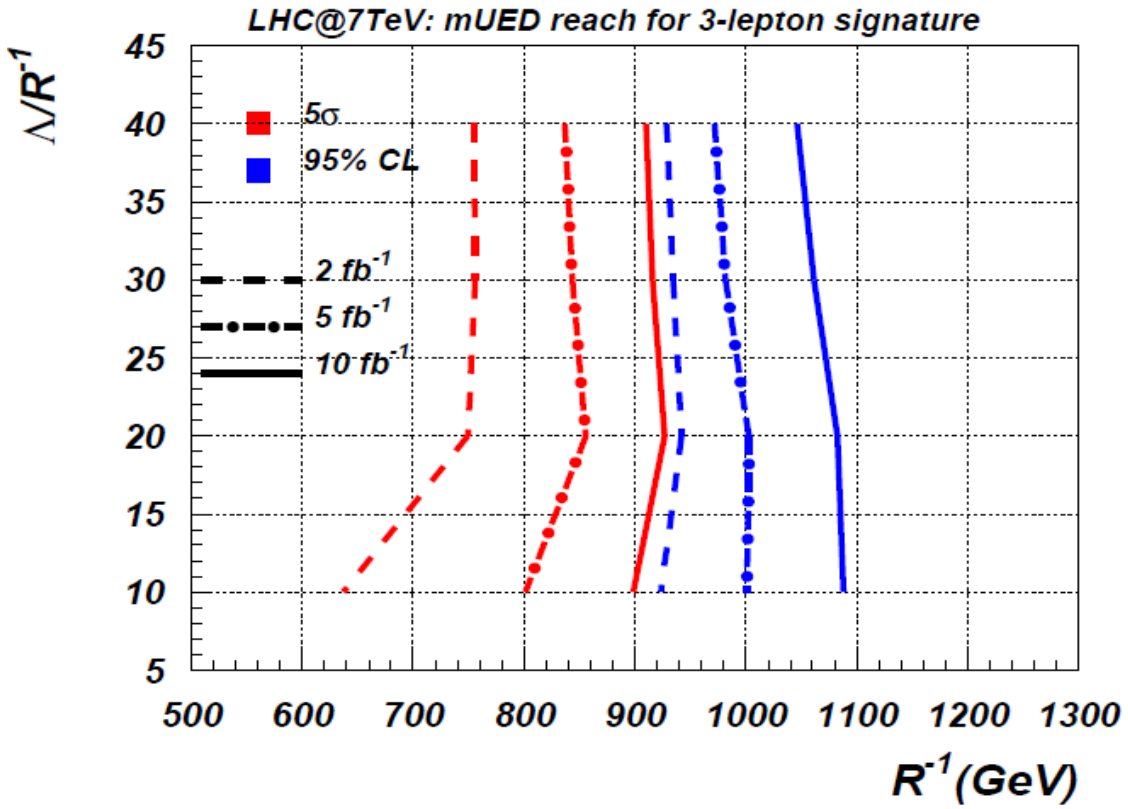
	Finished	Time(hr)
Symbolic	6498/6498	0.00
σ	4/4	3.29
Events	4/4	7.30

Results from calchep_batch at HEPMDB

CalcHEP Events Library

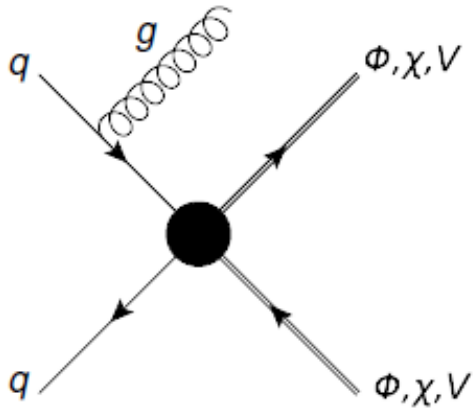
Home
Symbolic Results
Numerical Results
Events Library
Process Library

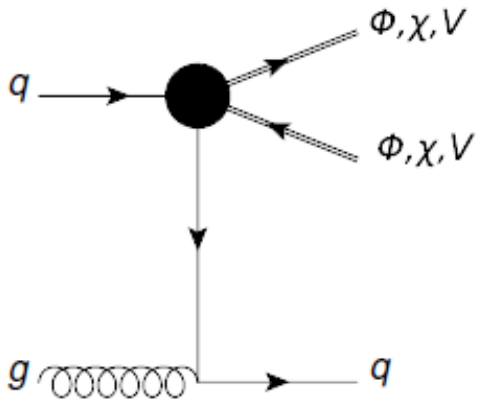
Date	LHE	plain Ntuple
Tue Mar 27 23:06:39 2012	Q1Q1_MH120_8tev-invR1000LR40.lhe	
Wed Mar 28 00:32:40 2012	Q1Q1_MH120_8tev-invR1200LR40.lhe	
Tue Mar 27 19:42:27 2012	Q1Q1_MH120_8tev-invR600LR40.lhe	
Tue Mar 27 21:34:29 2012	Q1Q1_MH120_8tev-invR800LR40.lhe	

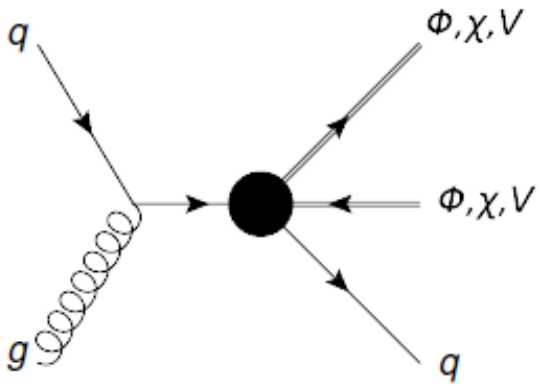


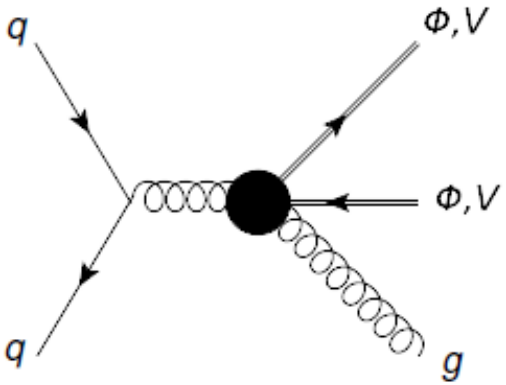
Complete DIM 5/6 DM operators study

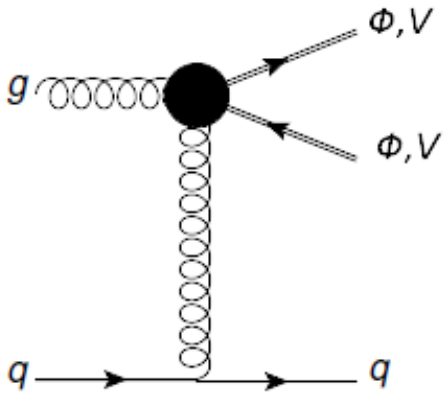
ArXiv:**1610.07545** L. Panizzi, A. Pukhov, M.Thomas, AB

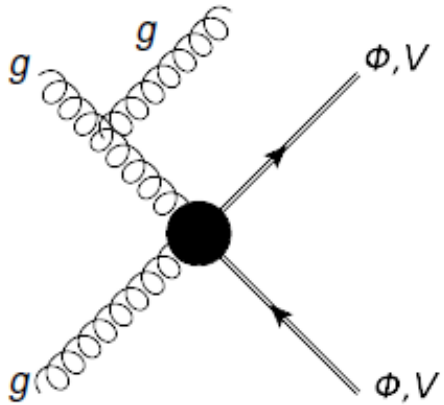


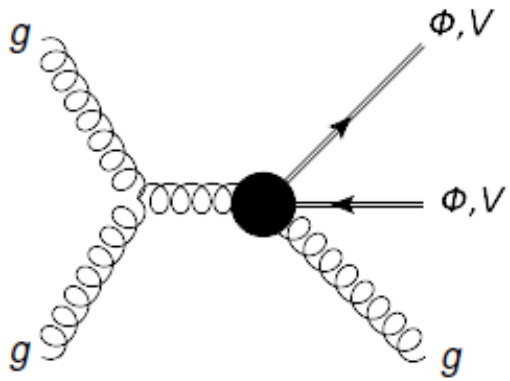


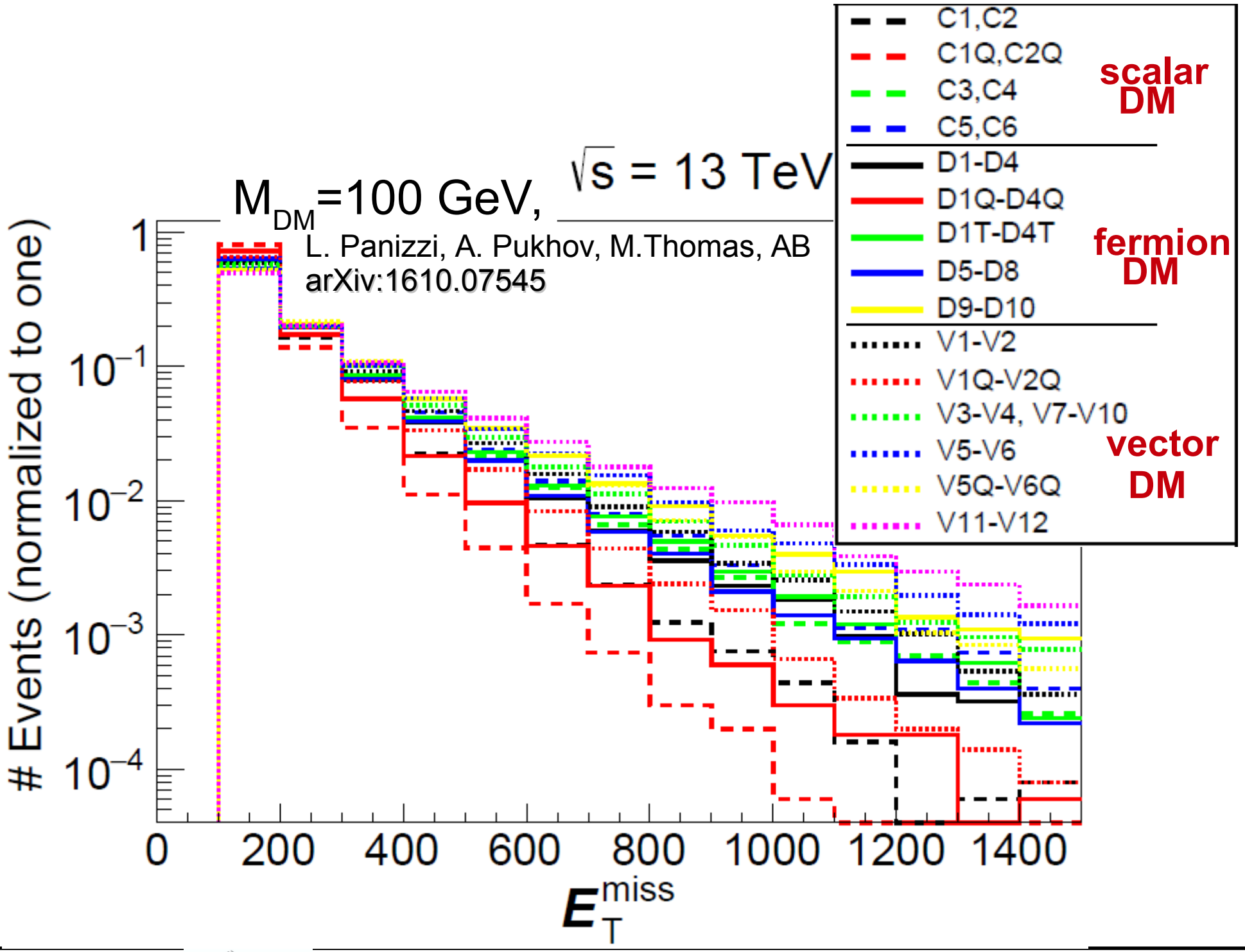











M_{DM}=100 GeV, $\sqrt{s}$ = 13 TeV
L. Panizzi, A. Pukhov, M.Thomas, AB
arXiv:1610.07545
Events (normalized to one)
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
200
400
600
800
1000
1200
1400
E_T^{miss}
C1,C2
C1Q,C2Q
C3,C4
C5,C6
scalar DM
D1-D4
D1Q-D4Q
D1T-D4T
D5-D8
D9-D10
fermion DM
V1-V2
V1Q-V2Q
V3-V4, V7-V10
V5-V6
V5Q-V6Q
V11-V12
vector DM

Additional CalcHEP Features

User routines

- CalcHEP as a matrix element generator for other packages examples/templates are in the **$CALCHEP/utiles/main_22.c**

$CALCHEP/bin/make_main [-o<exe_name>] <C source codes and libraries>

- **user-defined form-factor**

```
extern double usrFF(int nIn, int nOut,  double * pvect,char**pName,int*pCode);
/*
The usrFF function appears as factor at squared matrix element for
Monte Carlo calculations.  CalcHEP code has a dummy version of this function
which always return 1.  The dummy version is replaced on the  user one if
its code is passed to CalcHEP linker via 'Libraries' model file. One can use
CALCHEP and  WORK  environment variables  to specify path to the code.
These variables are  automatically defined in calchep and calchep_batch  scripts.
Also one can use any other environment variables defined separately.
 Parameters of usrFF:
    nIn - number of incoming particles;
    nOut- number of outgoing particles;
    pvect  presents momenta of particles:
       4-momentum of  i^{th} particle ( i=0,1,...,nIn+nOut-1)  is
       q[k]=pvect[4*i+k]  k=0,1,2,3;
       q[0] - in particle energy, which always is positive.
       q[3] - specify projection of momentum on axis of collision.
```

User routines

- **user-defined cuts**
  $CALCHEP/utile/usrfun.c

```
// Example: UMT(p1,p2) function which calculates transfer mass of 2 particles,
// for instance    UMT(e,Ne) - gives transverce mass of electron and neutrino.

double usrfun(char * name, int nIn, int nOut, double * pvect,char**pName,int*pCode)
{    char p1[10],p2[10];    // for 2 particles in MT(p1,p2)
     int i,j;
     double sum=0;

     if(name==strstr(name,"MT("))    // name is started from "MT("
     {  //read p1&p2
        int np=sscanf(name+3,"%[^,],%[^)]",p1,p2);
        for(i=nIn;i<nIn+nOut;i++)
        {  if(strcmp(p1,p2)==0) j=i+1;     /* if  p1==p2 */ else j=nIn;
           for(  ;j<nIn+nOut;j++)
           if(strcmp(p1,pName[i])==0 &&  strcmp(p2,pName[j])==0)
           //find position of particles
           { double * q1=pvect+4*i, *q2=pvect+4*j;
             double Et1=sqrt(fabs(q1[0]*q1[0] - q1[3]*q1[3]) );
             // transvers energy of the first particle
             double Et2=sqrt(fabs(q2[0]*q2[0] - q2[3]*q2[3]) );
             // transvers energy of the second particle
             sum+=sqrt( (Et1+Et2)*(Et1+Et2) -(q1[1]+q2[1])*(q1[1]+q2[1]) - (q1[
); // sqrt(E^2-PL^2)
           }
        }
     }  else { printf("Not defined user function %s\n",name); exit(2);}

     return sum;
}
```

User routines

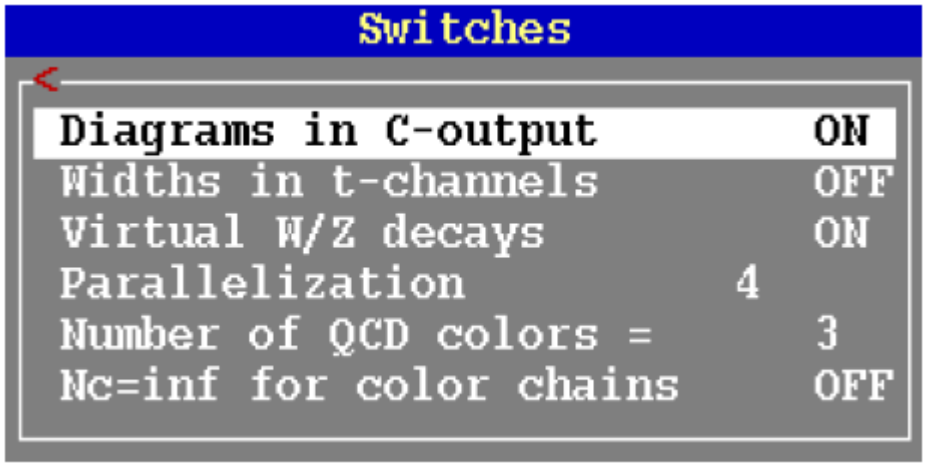
- **user-defined propagator**
  (alteration of the existing propagators)
  $CALCHEP/c_source/num/sqme_aux.c

```
Q1[i]=dmass[i]*dmass[i]-sqrMom(nin,Qtxt[i],momenta);
if(dwidth[i])
{  REAL w,w2, q2=Q1[i]*Q1[i];
    w=dmass[i]*dwidth[i];
    w2=w*w;
    if(q2>BWrange2*w2) {if(q2<(BWrange2+1)*w2) q2=(BWrange2+1) *w2; w2=0; }
    Q2[i]=1/(q2+w2);
    Q0[i]=Q2[i]*Q1[i]*Q1[i];
    Q1[i]*=Q2[i];
} else
{   if((Q1[i]>0? Q1[i]:-Q1[i]) < 10*s0max) err=2;
    if(!Q1[i]) Q1[i]=s0max;
    Q1[i]=1/Q1[i];
    Q2[i]=Q1[i]*Q1[i];
    Q0[i]=1;
}
}
return err;
```

New/recent features of CalcHEP

- ***Parallel calculations***

Program	Method
symbolic calculation of diagram	fork
writing of C-code	fork
compilation of C-code	not implemented
Vegas MC integration	threads
Generation of unweighted events	threads



New/recent features of CalcHEP

- ***Parallel calculations in batch regime***
  - ***PBS mode***

```
Parallelization method:      pbs
Walltime : 5
Memory   : 4
email:       name@address
Max number of processes per node: 16
Max number of cpus:       10
```

  - ***local mode***

```
Parallelization method:  local
Max number of processes per node: 1
Max number of cpus:       16
```

New/recent features of CalcHEP

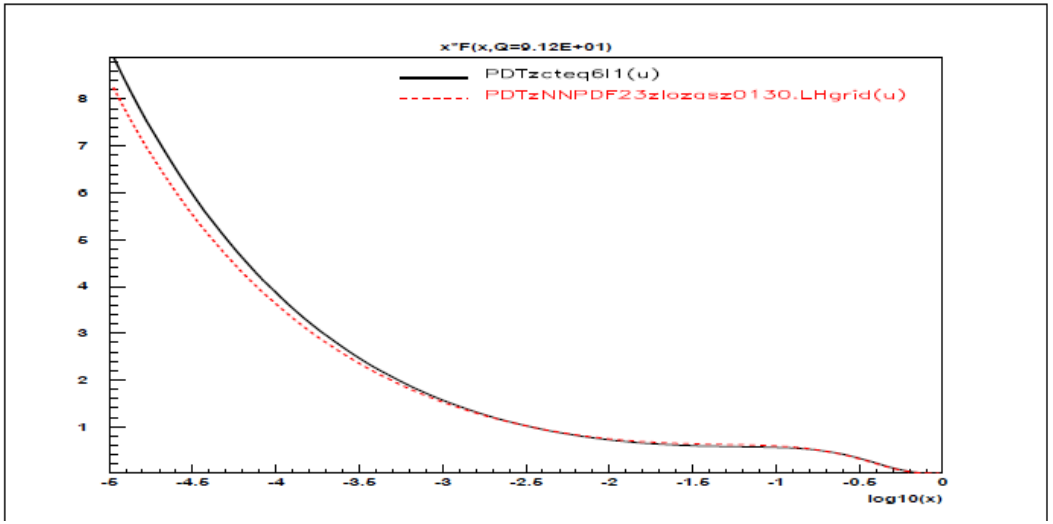
- ***New colour particles and vertices***

P1	P2	P3	color structure
3_a	$\bar{3}^b$		δ^a_b
6_{ab}	$\bar{6}^{cd}$		$(\delta^c_a\delta^d_b + \delta^d_a\delta^c_b)/2$
8_α	8_β		$\delta^{\alpha\beta}$
3_a	3_b	3_c	ϵ^{abc}
$\bar{3}^a$	$\bar{3}^b$	$\bar{3}^c$	$\bar{\epsilon}_{abc}$
8_α	8_β	8_γ	$-if^{\alpha\beta\gamma}$
3_a	$\bar{3}_b$	8_γ	$\tau^{\gamma\,a}_{\ b}$
6_{ab}	$\bar{6}^{cd}$	8_γ	$(\tau^{\gamma\,a}_{\ c}\delta^b_d + \tau^{\gamma\,a}_{\ d}\delta^b_c + \tau^{\gamma\,b}_{\ d}\delta^a_c + \tau^{\gamma\,b}_{\ c}\delta^a_d)/2$
6_{ab}	$\bar{3}^c$	$\bar{3}^d$	$(\delta^a_c\delta^b_d + \delta^a_d\delta^b_c)/2$
$\bar{6}^{ab}$	3_c	3_d	$(\delta^c_a\delta^d_b + \delta^d_a\delta^c_b)/2$

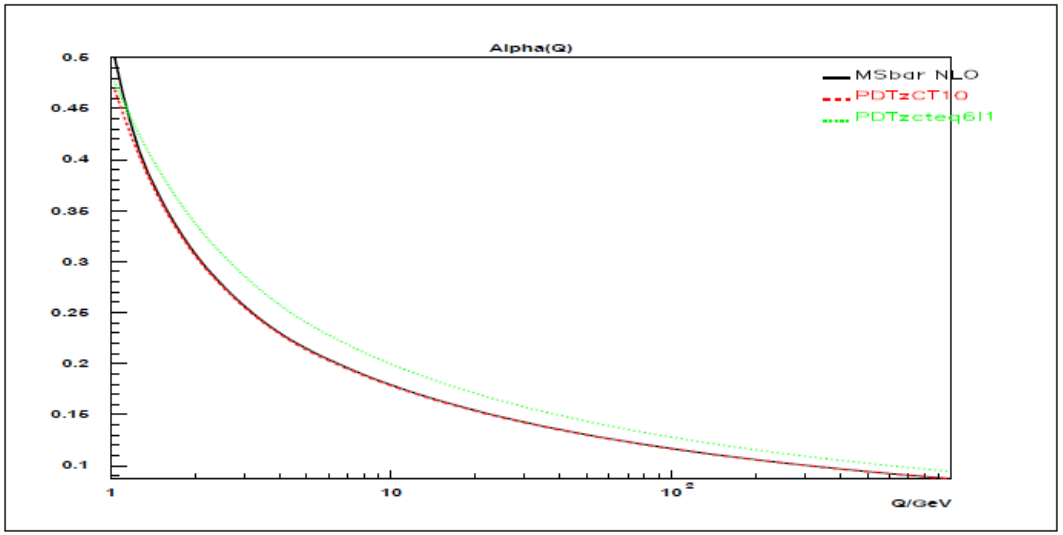
New/recent features of CalcHEP

- ***PDFs and visualisation***

```
PDF plots
x-Min = 1.000E-05
x-Max = 1.000E+00
q-Min = 1.500E+00
q-Max = 1.000E+04
Npoints = 100
q0      = 91.19GeV
x0      = 1.00E-01
log scale argument ON
Display plot x*F(x)
Display plot   F(x)
Display plot   F(Q)
both PDF1&PDF2    ON
```




```
QCD alpha
parton dist. alpha pdf1
alpha(MZ)=  0.1184
nf =        5
order=      NLO
mb(mb)=     4.200
Mtop(pole)= 173.00
Alpha(Q) plot
Qren = M12
Qpdf1= Qren
Qpdf2= Qpdf1
Qshow= Qren
```



lhe→pythia8→delphes→root

- ***This can be done in automatically using*** **CheckMate2**

  - No intermediate HEPMC files
  - PYTHIA8, can control it via cards
  - Delphes3, can control it via cards
  - Produces root files
  - Can use CM statistical analysis routines and check the signal exclusio
  - One can produce muldi-dim scan at HEPMDB and direct lhe files to CM2

lhe→pythia8→delphes→root

./CheckMATE lhe-pythia8-delphes.dat
where lhe-pythia8-delphes.dat contains lines

```
[Parameters]
Name: lhe-pythia8-delphes
SkipAnalysis: True
SkipParamCheck: True
WriteDelphesEvents: True
OutputExists: Overwrite
[myprocess]
Pythia8Card: testpythia8card.dat
```

And testpythia8card.in

```
! Settings used in the main program.
Next:numberCount = 500            ! print message every n events
Next:numberShowInfo = 2           ! print event information n times
Next:numberShowProcess = 2          ! print process record n times
Next:numberShowEvent = 2          ! print event record n times
Main:numberOfEvents = 200         ! number of events to generate
Beams:frameType = 4               ! use LHEF input
Beams:LHEF = =lhe_from_calchep.lhe !  LHEF events file
```

Final remarks

- use launchpad to file problems or ask questions – answers will be available to everybody!
- read manuals – they have much more details
- tools are powerful but
  should not be blindly trusted or blamed!
- use independent programs to for double check, use limits to check if your results make sense
- Let me know if you are interested in CalcHEP/HEPMDB tutorial!

BACKUP

Compilation, potential problem and its solution

- **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

Compilation, potential problem and its solution

- **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

- **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"

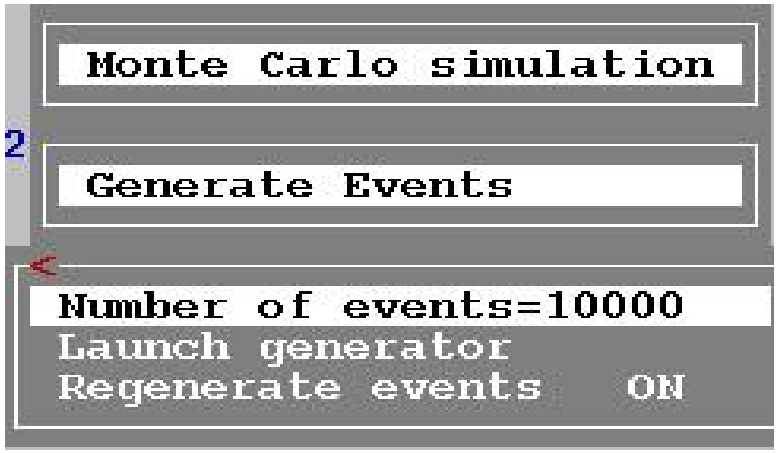
Compilation, potential problem and its solution

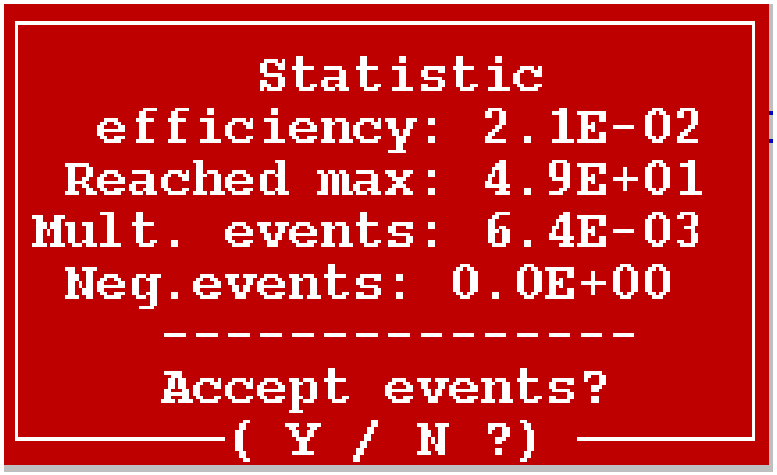
- To compile the CalcHEP source code you need:
  C compiler, the X11 graphics library and the X11 include files
  "CalcHEP is compiled successfully and can be started "
  is a good sign

- 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"
- Potential problem in compilation
  - The most frequent compilation problem is due to the absence of the X11 include files; CalcHEP still compiles, 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 install to run CalcHEP in GUI mode
    - libX11-devel for Fedora/Scientific, Darwin(MAC)
    - libX11-dev for Ubuntu/Debian
    - xorg-x11-devel for SUSE

events generations





File with events in the native CalcHEP format

events_1.txt - /home/belyaev/Dropbox/hep_tools/calchep/calc_work/pp_Wbb_example/

File Edit Search Preferences Shell Macro Windows Help

/home/belyaev/Dropbox/hep_tools/calchep/calc_work/pp_Wbb_example/events_1.txt 243603 bytes L: 1 C: 0

```
#CalcHEP  3.4.cpc
#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
```

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 above became a part of calchep_batch interface – will be discussed below

protocol prt_*nn*

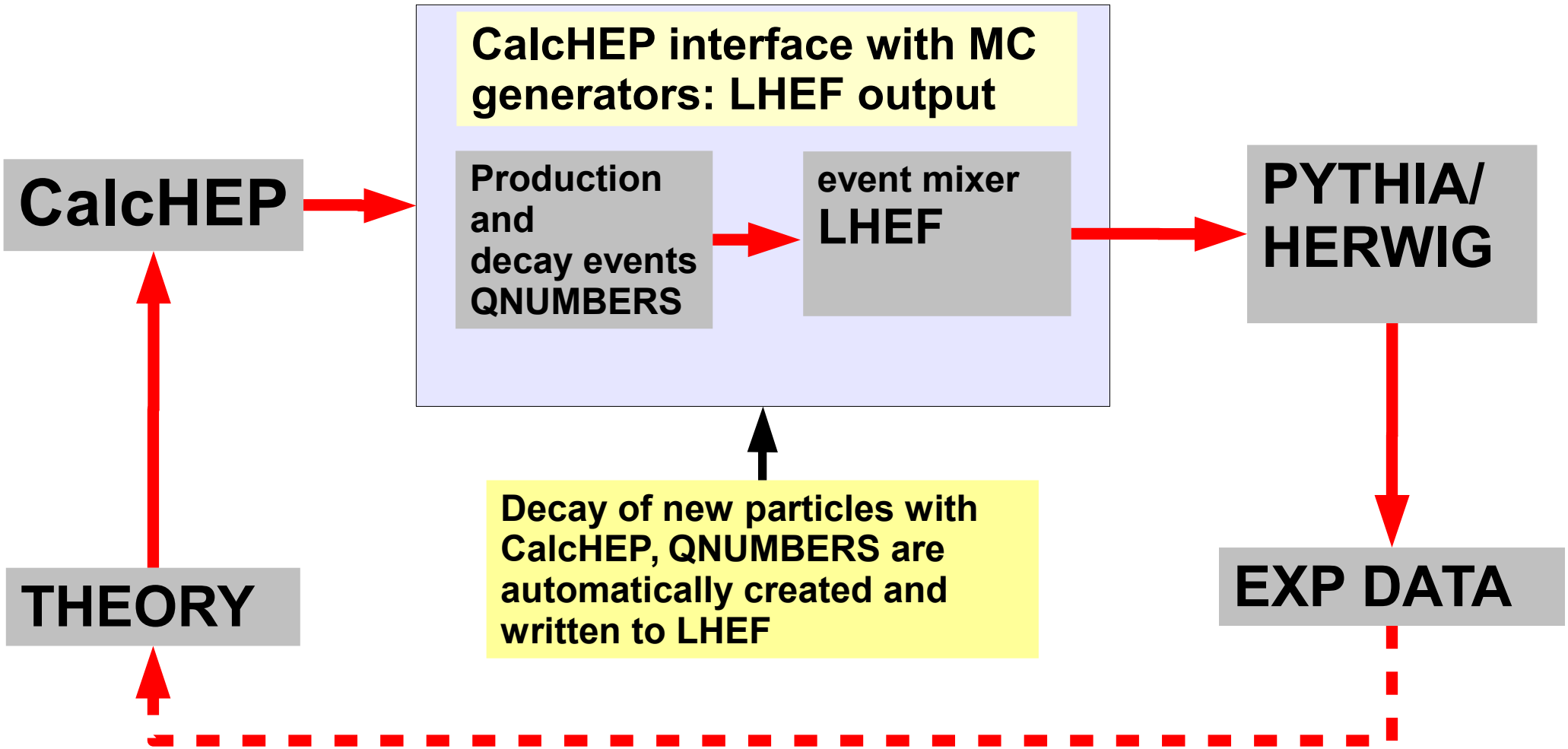
```
    CalcHEP kinematics module
 The session parameters:

#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 = 7.8180609999999E-03
    alfSMZ = 1.1720000000000E-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
.........................................
.........................................
```

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

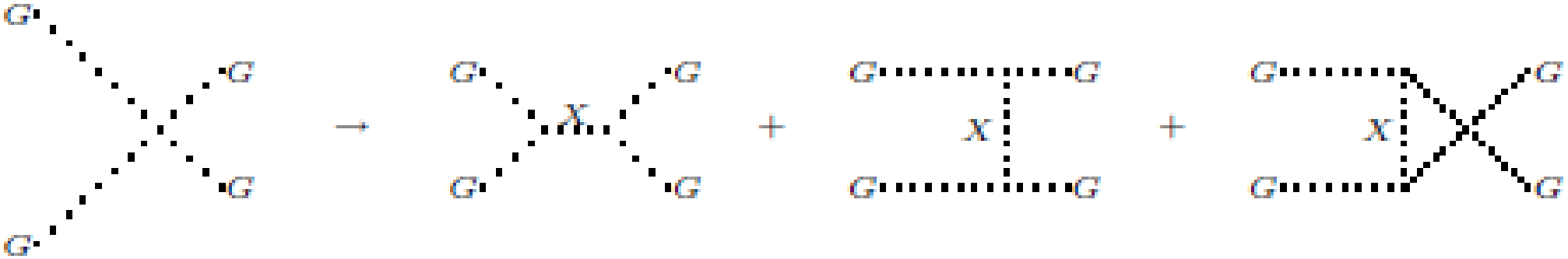
  p1: 1000*rtS/2
  p2: 1000*rtS/2
  rtS here is some “fake” parameter
- you can use “fake” parameter only if you define it as a loop parameter
  - It can be used in the cut statement (assigning cut to the symbol)
  - It can be assigned to the parameter model – this way you can run use complicated scan

CalcHEP interface to MC generators via Events in the LHE format



Vertices with color 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 color 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_'*
  ('_' is the number of the particle in table item).

Vertices

Clr	Del	Size	Read	ErrMes			
A1	|A2	|A3	|A4	|>	Factor	<|>	Lorentz part
G	|G	|G	|	|GG			|m1.m2*(p1-p2).m3+m2.m3*(p2-p3).m1+m3.m1*(p3-p1).m2
G	|G	|G.t	|	|GG/Sqrt2			|m1.M3*m2.m3-m1.m3*m2.M3

Vertices with color particles in LanHEP

- *The splitting of vertex with 4 colored particle into 3-particles vertices is done by LanHEP automatically: each vertex containing 4 color 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*
- *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).*
- *the default value is 2 for SplitCol1 and 1 for SplitCol2*

Implementation of SM Lagrangian(1)

Location of LanHEP model files:

lanhep/mdl/stand.mdl

```
%
% Standard Model - unitary and t'Hooft-Feynman gauges.
%

keys gauge_fixing=Feynman.

do_if gauge_fixing==Feynman.
    model 'Stand. Model (Feyn. gauge)'/6.
do_else_if gauge_fixing==unitary.
    model 'Stand. Model (un. gauge)'/5.
do_else.
    write('Error: the key "gauge" should be either "Feynman" or "unitary".').
    quit.
end_if.
```

Implementation of SM Lagrangian(2)

- ***Parameters definition***

```
let g5=gamma5.
use sm_tex.

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 (PDG-94,"on-shell")',
           s12 = 0.221   : 'Parameter of C-K-M matrix (PDG-94)',
           s23 = 0.040   : 'Parameter of C-K-M matrix (PDG-94)',
           s13 = 0.0035  : 'Parameter of C-K-M matrix (PDG-94)'.

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

parameter  c12  = sqrt(1-s12**2) :      'parameter  of C-K-M matrix',
           c23  = sqrt(1-s23**2) :      'parameter  of C-K-M matrix',
           c13  = sqrt(1-s13**2) :      'parameter  of C-K-M matrix'.

parameter  Vud = c12*c13                    : 'C-K-M matrix element',
           Vus = s12*c13                    : 'C-K-M matrix element',
           Vub = s13                        : 'C-K-M matrix element',
           Vcd = (-s12*c23-c12*s23*s13)     : 'C-K-M matrix element',
           Vcs = (c12*c23-s12*s23*s13)      : 'C-K-M matrix element',
           Vcb = s23*c13                    : 'C-K-M matrix element',
           Vtd = (s12*s23-c12*c23*s13)      : 'C-K-M matrix element',
           Vts = (-c12*s23-s12*c23*s13)     : 'C-K-M matrix element',
           Vtb = c23*c13                    : 'C-K-M matrix element'.

OrthMatrix( { {Vud,Vus,Vub}, {Vcd,Vcs,Vcb}, {Vtd,Vts,Vtb}} ).
```

Implementation of SM Lagrangian(4)

- ***Definition of mixings and doublets***

```
let l1={n1,e1}, L1={N1,E1}.
let l2={n2,e2}, L2={N2,E2}.
let l3={n3,e3}, L3={N3,E3}.

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}.

let B1= -SW*Z+CW*A, W3=CW*Z+SW*A, W1=('W+'+'W-')/Sqrt2,
         W2 = i*('W+'-'W-')/Sqrt2.

do_if gauge_fixing==Feynman.

let gh1 = ('W+.c'+'W-.c')/Sqrt2, gh2= i*('W+.c'-'W-.c')/Sqrt2,
                gh3= CW*'Z.c'+SW*'A.c', gh={gh1,gh2,gh3}.

let Gh1 = ('W+.C'+'W-.C')/Sqrt2, Gh2=i*('W+.C'-'W-.C')/Sqrt2,
                Gh3= CW*'Z.C'+SW*'A.C', Gh={Gh1,Gh2,Gh3}.

end_if.

let WW1 = {W1,  W2 , W3}, WW = {'W+',W3,'W-'}.

let g=EE/SW, g1=EE/CW.
```

LanHEP

- ***Index order***
  SetDefIndex(spinor, color c3, color c8, vector).
- ***Example: implementation of***

$$\mathcal{O}_{tW} = \bar{q}\sigma_{\mu\nu}\tau^i t\tilde{\phi}W_i^{\mu\nu}$$

interactions

- ***Let statements:***

  You should write explicitly all indices in the let statement or hide them all!

```
parameter ftW=0.
parameter Lam=1000.

let sigma^i^j^mu^nu=
i*(gamma^i^k^mu*gamma^k^j^nu - gamma^i^k^nu*gamma^k^j^mu)/2.
let phitilde = i*tau2*PP.
```

LanHEP

- tau indicies are not in the default order, so they should be shown explicitly

$$\mathcal{O}_{tW} = \bar{q}\sigma_{\mu\nu}\tau^i t\tilde{\phi}W_i^{\mu\nu}$$

SetDefIndex(spinor, color c3, color c8, vector).

```
lterm ftW/Lam**2*(Q3^i*sigma^mu^nu*tau^i^j^a*t)*phitilde^j*F^mu^nu^a
where
F^mu^nu^a=deriv^mu*WW1^nu^a-deriv^nu*WW1^mu^a
+ AddHermConj.
```

- or, alternatively one can add index 2 in the default order and write **lterm** in compact way

```
SetDefIndex(spinor, color c3, color c8, vector, wild 2).

lterm ftW/Lam**2*Q3*sigma^mu^nu*tau^a*phitilde*t*F^a^mu^nu
+ AddHermConj
where
F^a^mu^nu=deriv^mu*WW1^nu^a-deriv^nu*WW1^mu^a.
```

PhenoData

- spin-off the **PhenoData**
  hepmdb.soton.ac.uk/phenodata
- stores data (digitized curves from figures, tables etc) from those HEP papers which did not provide data in arXiv or HEPData, and to avoid duplication of work of HEP researchers on digitizing plots.
- has an easy search interface and paper identification via arXiv, DOI or preprint numbers. PhenoData is not intended to be a replication of any existing archive
- Has batch upload mode

CalcHEP

scan_example.c

```
#include<math.h>
#include<stdio.h>
#include<unistd.h>
#include<sys/stat.h>
#include<sys/types.h>
#include <dlfcn.h>
#include <sys/wait.h>
#include"num_in.h"
#include"num_out.h"
#include"VandP.h"
#include"dynamic_cs.h"
#include"rootDir.h"
#include <time.h>

int main(void)
{ int err,i;

          /* INTPUT PARAMETERS (to scan over) */
      double Mh,   Mhmin=110,       Mhmax=150;

          /* OUTPUT PARAMETERS */
      // Higgs decay branching ratios
      double  wh,braa;
txtList branchings;

//set model dir here
char mdldir[] = "models";

 // Set model number and number of points to collect, mdlnr is your model number
int mdlnr=3, npoints=50;

//a model to switch between to reset values when reloading
 setModel(mdldir , mdlnr );
```

CalcHEP

scan_example.c

```
/*****************************************************************************/
 srand (time(NULL)); //this is used to seed the random number by the system time

 if (remove("scan.dat") == -1)
        perror("Error in deleting a file");

 FILE *file;
 file = fopen("scan.dat","a+"); /* apend file (add text to
                                   a file or create a file if it does not exist.*/
 // Writing parameter names at first line to keep track of columns:
 //input parameters (1)
 //output parameters (3)
 fprintf(file,"Mh\t\twh\t\tbraa\n");
 fclose(file); /*done with header of file*/

 /*** Starting randomizing loop ***/
 for (i = 1; i <= npoints; i++){

 /********** generate random values for variables **********/
 Mh      = Mhmin+(double) random()/RAND_MAX*(Mhmax-Mhmin);

 /* Have to reset model every time, otherwise widths are not recalculated */
 setModel(mdldir , mdlnr );

 /******* assign variable values ********/
 /* the string is the calchep var name */
 err=assignValW("Mh", Mh);

 // Calculation of public constraints
 err=calcMainFunc();
```

CalcHEP

scan_example.c

```
 if(err!=0) {
          printf("Can not calculate constrained parameter
%s\n",varNames[err]);i--;
 }
 else {
                  // if the point survives the constraints collect more output
values:
                  // width and branchings of a particle
                  wh      = pWidth("h",&branchings);
                  braa     = findBr(branchings,"A,A");

                  // write values to file
                  file  = fopen("scan.dat","a+");
                  //input parameters
                  fprintf(file,"%f\t",Mh);
                  //output parameters
                  fprintf(file,"%f\t%e\n",wh,braa);
                  fclose(file);
 }

 }// *** end of rand loop ***

   return 0;
}
```

CalcHEP

$CALCHEP/bin/make_main scan_example.c
a.out
more scan.dat

Mh	wh	braa
135.996838	0.006698	2.099067e-03
116.973931	0.003420	2.160684e-03
132.554627	0.005675	2.198545e-03
127.711034	0.004660	2.271069e-03
130.134697	0.005117	2.244877e-03
115.663777	0.003326	2.126571e-03
111.244676	0.003048	2.000407e-03
139.144130	0.007952	1.977188e-03
123.835785	0.004091	2.271298e-03
139.866680	0.008296	1.945347e-03
112.663815	0.003139	2.037072e-03
123.800804	0.004087	2.271076e-03

- **One can perform a powerful scan of parameter space, Br's, cross sections, ...**