

CalcHEP: Tools for event generation and other new facilities of the package.

A.Pukhov, SINP MSU, Russia

***International Linear Collider (ILC) Workshop
Valencia, 6-10 November 2006***

Short introduction in CalcHEP.

CalcHEP is a package for computation of Feynman diagrams, integration over multi-particle phase space and event generation for generic model.

Computation consists of two steps, symbolic and numerical ones. At symbolic step CalcHEP calculates symbolic expressions for squared Feynman diagrams and transforms them into C-code. This C-code is compiled and linked with routines which perform Monte Carlo integration and event generation.

There are two versions of this soft

CompHEP: <http://theory.sinp.msu.ru/comphep>

presented by Boos, Dubinin, Bunichev, Dudko, Edneral, Kryukov, Ilyin, Savrin, Sherstnev, Semenov.

and CalcHEP: <http://theory.sinp.msu.ru/~pukhov/calchep.html>

presented by me.

In general both versions have the same facilities, but some details after 5 years of independent development are different.

Short introduction: the models implemented.

In CalcHEP there are implemented

- *The Standard Model,*
- *MSSM based on SuSpect/Isajet/SoftSUSY/SPHENO (MicrOMEGAs group)*
- *NMSSM based NMHDecay (MicrOMEGAs group + C.Hugonie)*
- *MSSM with CP violation based on CPsuperHiggs(MicrOMEGAs + S.Kraml)*
- *The Little Higgs model (A.Birkedal et al; A.Belyaev et al)*
- *The extra dimension LQP model (G. Servant, K. Agashe)*
- *The LeptoQuark model (A.Belyaev et al)*

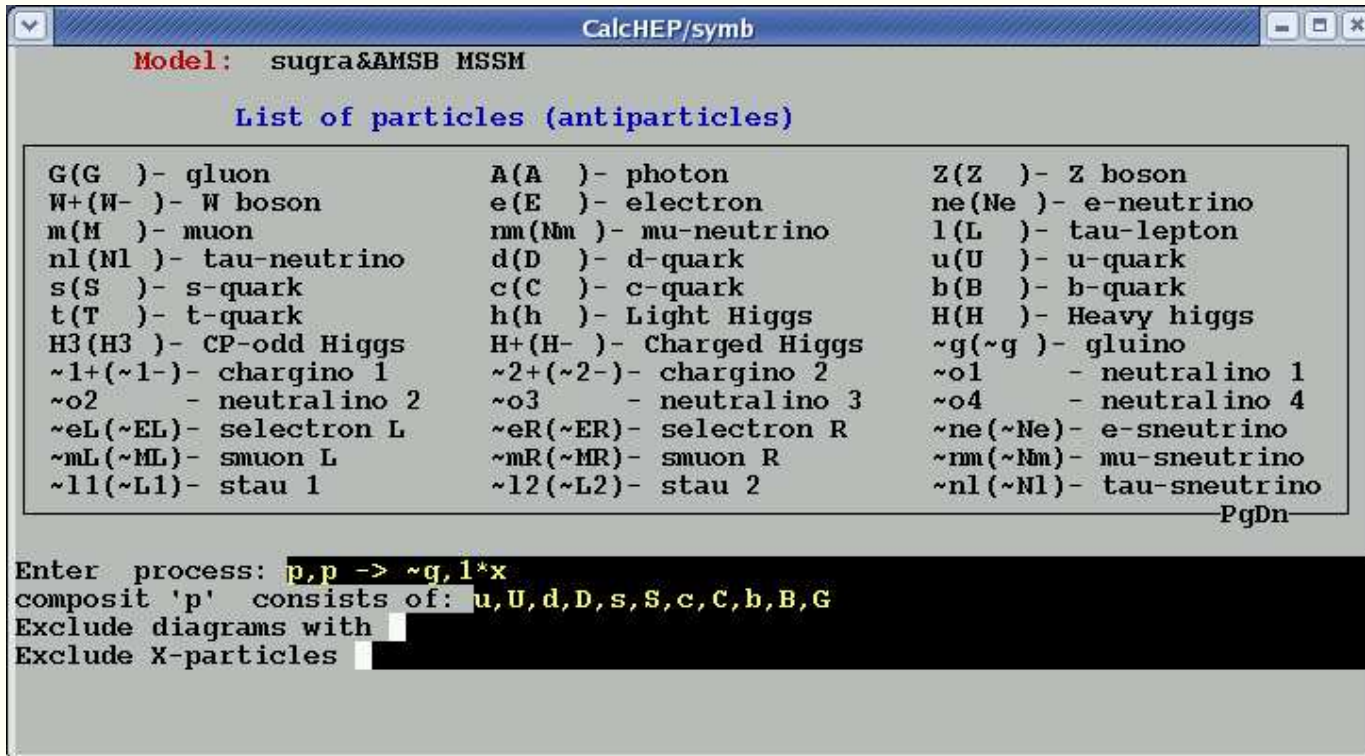
Implementation of new model is simplified by LanHEP package

A. Semenov. Nucl.Inst.&Meth. A393 (1997) p. 293.

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

Short introduction: Enter of Process.

CalcHEP is a program with user-friendly interface.



```
CalcHEP/symb
Model: sugra&AMSB MSSM

List of particles (antiparticles)

G(G )- gluon
W+(W- )- W boson
m(M )- muon
nl(Nl )- tau-neutrino
s(S )- s-quark
t(T )- t-quark
H3(H3 )- CP-odd Higgs
~1+(~1-)- chargino 1
~o2 - neutralino 2
~eL(~EL)- selectron L
~mL(~ML)- smuon L
~l1(~L1)- stau 1

A(A )- photon
e(E )- electron
nm(Nm )- mu-neutrino
d(D )- d-quark
c(C )- c-quark
h(h )- Light Higgs
H+(H- )- Charged Higgs
~2+(~2-)- chargino 2
~o3 - neutralino 3
~eR(~ER)- selectron R
~mR(~MR)- smuon R
~l2(~L2)- stau 2

Z(Z )- Z boson
ne(Ne )- e-neutrino
l(L )- tau-lepton
u(U )- u-quark
b(B )- b-quark
H(H )- Heavy higgs
~g(~g )- gluino
~o1 - neutralino 1
~o4 - neutralino 4
~ne(~Ne)- e-sneutrino
~nm(~Nm)- mu-sneutrino
~nl(~Nl)- tau-sneutrino

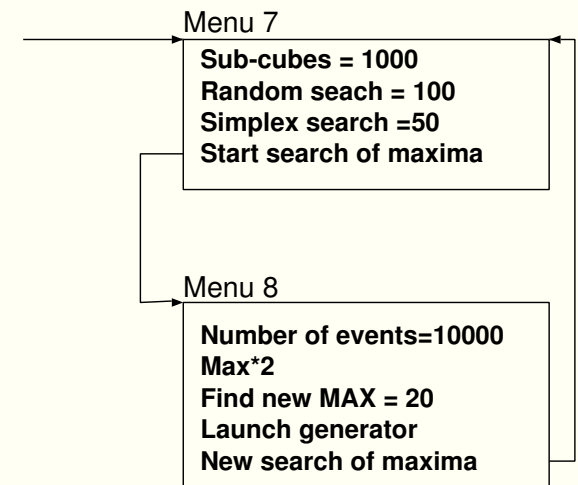
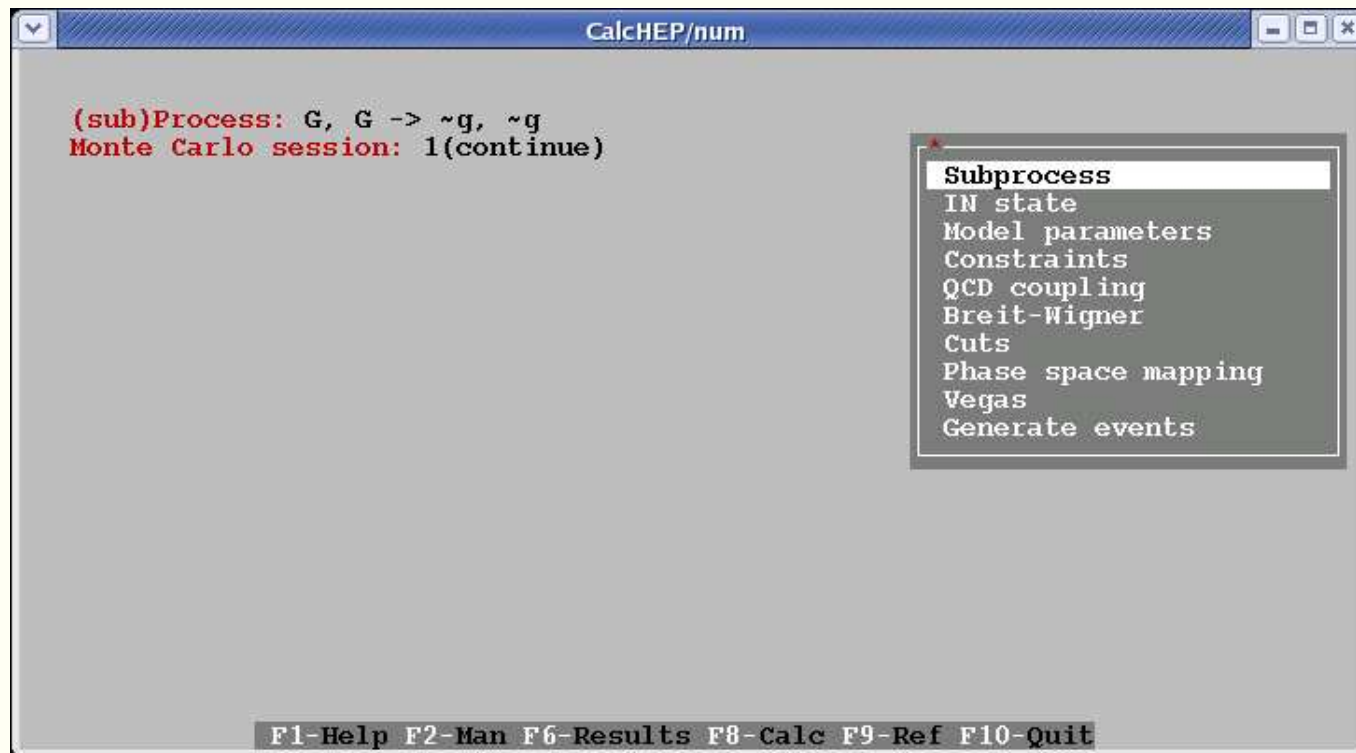
PgDn

Enter process: p,p -> ~g,1*x
composit 'p' consists of: u,U,d,D,s,S,c,C,b,B,G
Exclude diagrams with 
Exclude X-particles
```

Short introduction. Numerical session.

CalcHEP is a menu driven system with Help option.

Press the F1 key to get information about current menu position.



Short introduction: Batch calculations

Batch non-interactive session can be realized by the following trick:

In the end of interactive session launched with option "+blind", for example,

```
n_calchep +blind
```

program writes on the screen the sequence of pressed keys. Let we have generated 123 events using CalcHEP menus. Then after the end of session we find on screen:

```
"[[[[[[[[[[[[{[[[[{{123{[[[[{0"
```

In principle, one can decode it, { - Enter ; [- Down arrow

To force CalcHEP to repeat this session one can call

```
n_calchep -blind "[[[[[[[[[[[[[{[[[[{{123{[[[[{0"
```

Now one can replace "123" to \$1 and save this line in a file to get a shell script for event generation.

Several such scripts are written and disposed in \$CALCHEP\bin.

Event generation in CalcHEP_2.5

CalcHEP_2.5 - is a new version which is in progress. Beta release of it is presented on CalcHEP site. It contains a script for generation of events in cycle over subprocesses.

```
$CALCHEP/bin/subproc_cycle Luminosity/nEvens [nMax]
```

The first parameter in case of collision means luminosity in (fb), but in case of decays it defines the number of decay events for each particle. The second parameter limits the number of events generated in collision processes.

In case of collisions program calculates cross section for each channel, after that it estimates the needed number of events and generates them.

In case of decay processes, the program calculates all partial widths for a given particle and after that on the second cycle generates for each channel the number of events proportion to its partial width.

Interface with Showering and Hadronization event Generators

This interface is based on Les Houches Accord for event generators

Boos et al, hep-ph/0109068. Here Fortran interface is assumed and information is passed via COMMON BLOCKs.

CalcHEP has two interface program files stored in directory \$CALCHEP/utile. One of them callPYTH.f is a Fortran main, the second one event2pyth2.c contains auxiliary routines. They both have to be compiled and linked with PYTHIA. (After small corrections it should work with other SHGs.)

In result of subproc_cycle or generation of events in interactive sessions we get several files event_#.txt. The

```
CALL SCANDIR('dirname')
```

opens all event_#.txt files and read their headers. It can be called several times for different directories.

After that any call of UPEVNT will choose a subprocess according to its cross section, read an event record and fill interface COMMON BLOCKs.

Particle decays and Double Les Houches Accord.

Events with new particles passed to SHG have to be accompanied with information about decays. For instance, one can use SUSY Les Houches Accord (P. Skands et al, hep-ph/0311123). Here information is passed via text file using PDG numbers for particle identification.

```
DECAY 1000037 3.895221E+00 # second chargino(~2+) width
# Branching Nout ID_1 ID_2 ....
1.8877E-03 2 24 1000022 # Br(~2+ -> W+ ~o1)
2.4367E-01 2 25 1000024 # Br(~2+ -> h ~1+)
.....
```

Initially this tool was designed for MSSM. In order to extend it to generic model one has to get a possibility to register new particle in SHG programs. In case of PYTHIA it is done in the SLHA style:

```
BLOCK QNUMBERS 1000045 # ~chi_50
1 0 # 3 times electric charge
2 2 # number of spin states (2S+1)
3 1 # color rep (1: singlet, 3: triplet, 8: octet)
4 0 # Particle/Antiparticle distinction (0=own anti)
```

CalcHEP has an option to generate SLHA decay file, but it takes into account only 1 → 2 decays.

Fusion of collision and decay events in CalcHEP.

Apart of well known problem of lost of spin correlation the passing decay information via SLHA gets a flat momenta distributions and has a problem with color rearrangement. For example, ISAJET generates t -quark decays

DECAY	6	1.21561825E+00	# TP	decays								
#	BR	NDA	ID1	ID2	ID3	ID4						
	3.33333343E-01	3	2	-1	5		# TP	-->	UP	DB	BT	
	3.33333343E-01	3	4	-3	5		# TP	-->	CH	SB	BT	
	1.11111112E-01	3	-11	12	5		# TP	-->	E+	NUE	BT	
	1.11111112E-01	3	-13	14	5		# TP	-->	MU+	NUM	BT	
	1.11111112E-01	3	-15	16	5		# TP	-->	TAU+	NUT	BT	

which leads to a crash in PYTHIA at the color rearrangement step. Also it will produce wrong mass distribution for $f\bar{f}$ pairs.

If parton events are passed to SHG by external program, the same program should be able to generate decay events and implement them decays before passing event to SHG.

In CalcHEP_2.5 this idea is realized. The same

```
CALL SCANDIR('dirname')
```

command reads decay events and applies them to outgoing particles.

Other new facilities.

Polarized beams. There is a possibility to calculate matrix element for polarized massless incoming particles. Polarized particle should be marked by the % symbol when we define the process.

$$E\%, e\% \rightarrow 2 * x \quad \text{or} \quad A\%, A\% \rightarrow 2 * x$$

In numerical session one can specify left/right contents of incoming beams.

Spin 3/2 and spin 2 massive particles are available in CalcHEP now. Spin 2 can be interesting for models with extra dimensions.

2 -parameter plots for parton distribution calculated by CalcHEP during Monte Carlo integration are available. There is an option to view plots generated in CalcHEP by PAW and Gnuplot. For such a goal CalcHEP files contains a code which should be moved to PAW or Gnuplot input file.

```
#--- GNUPLOT section ---
#GNUPLOT set title 'E,e ->m,M'
#GNUPLOT set xlabel 'cos(p1,p3)'
#GNUPLOT set ylabel 'Diff. cross section [pb]'
#GNUPLOT plot[-1:1] 'plot_1.txt' using (-1 +$0*0.02):1 w l

#--- PAW section ---
#PAW TITLE 'E,e ->m,M'
#PAW vector/Create X1(101)
#PAW sigma X1=ARRAY(101,-1#1)
#PAW vector/Create Y1(101)
#PAW vector/Read Y1 'plot_1.txt' ' ' 'OC
#PAW GRAPH 101 X1 Y1
```

Future development.

To complete version 2.5 I am going to design cuts and QCD scale definition which can be applied to all subprocesses simultaneously.

There is request to modify laser γ distribution function which should be sensitive to polarizations of incoming electron beam and γ .

It should be very useful to implement polarization for massive particles. I hope it will work for small legs in spite of squared diagram method.

*There is an idea to swap symbolic and numerical sessions. Now even for $1 \rightarrow 2 * x$ CalcHEP generates too many subprocesses which indeed are closed because of mass relations. Numerical CalcHEP will send request to symbolic one to generate list of available subprocesses. But request to generate codes for reactions will be sent only for subprocesses that are permitted by mass relations. The new generated codes will be linked dynamically to the `n_calchep`. This idea actually was realized in micrOMEGAs package. Here co-annihilation channels for calculation of relic density are compiled and attached to main code dynamically.*