

## **Recent development of CalcHEP**

**A.Pukhov, SINP MSU, Russia**

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

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

**CalcHEP starts from reading of model files which present Feynman rules for given model. It performs symbolic calculation of squared diagrams result of which can be presented in form of C code for subsequent numerical processing. This way gives a flexibility respect to new model realization.**

**Originally code of CalcHEP was identical to code of CompHEP presented here by E.Boos. Last years CalcHEP was developed mainly for astoparticles physics as a kernel micrOMEGAs package. Nevertheless its application for accelerator physics was significantly improved as well. This job was done in collaboration with Alexander Belyaev who will present here CalcHEP interface with MC generators in details.**

## Models with spin 2 and spin 3/2 particles.

*A possibility to work with high spin particles was realized*

Particles								
Full name	P	aP	number	2*spin	mass	width	color	aux
electron	e	E	11	1	0	0	1	
s_electron	$\tilde{e}_L$	$\tilde{E}_L$	1000011	0	MSeL	0	1	
Gravitino	$\tilde{g}$	$\tilde{g}$	1000039	3	MGr	wGr	1	
Massive Graviton	Gh	Gh	2000039	4	MHGr	whGr	1	

*Then interaction can be presented like*

Vertices							
A1	A2	A3	A4	>	Factor	<   >	Lorentz part
$\tilde{e}_L$	E	$\tilde{g}$			$i/\text{Sqrt}2/M_{pl}$		$G(m3)*G(p1)$
$\tilde{E}_L$	$\tilde{g}$	e			$i/\text{Sqrt}2/M_{pl}$		$G(p1)*G(m2)$

## ***Processes with polarized incoming particles.***

*It is realized for incoming massless particles of spin 1/2 and spin 1. One have to use the % symbol before particle name to force CalcHEP to calculate diagrams symbolically using polarized density matrix.*

$e\%, E\% \rightarrow 2*x$

*In numerical session the user can set numerical values for helicities.*

## Automatic width calculation.

*Particle widths included in the model can be calculated automatically using vertices of interaction. It mainly is interesting for BSM particles whose masses unknown.*

Particles									
Full name	P	aP	number	2*spin	mass	width	color	aux	
Light Higgs	h	h	25	0	Mh	!wh	1		
Heavy higgs	H	H	35	0	MHH	!wHh	1		

*The exclamation mark at particle width is a instruction to CalcHEP to generate the code for  $h \rightarrow 2*x$ , and  $H \rightarrow 2*x$  and compile them together with the codes of main processes.*

## **Batch mode.**

*Initially CalcHEP was designed for work in interactive sessions. To organize batch calculation the following trick is used. Interactive session launched with option "+blind",*

```
s_calchep +blind
```

*writes on the screen the sequence of pressed keys. Let we enter the process e,E->W+,W-, perform symbolic calculation and write C code for matrix elements. Then output should be*

```
{{e,E->W+,W-{{[[[[{{{0
```

*Here {,},[,] drives CalcHEP menus. Then*

```
s_calchep -blind "{{e,E->W+,W-{{[[[[{{{0"
```

*repeats previous session in non-interactive regime. CalcHEP output anyway is written into files. So, there is possibility to write a various scripts for batch calculation in CalcHEP without touching the main code.*

## **Cycle over subprocesses for cross sections/widths calculation and**

**There are several batch script commands presented in CalcHEP/bin directory. Launched without parameters they explain which parameters need.**

```
../../bin/subproc_cycle
```

```
Error, subproc_cycle in case of collision needs 1 obligatory parameter:
```

```
1: Integral luminosity in [1/fb] units
```

```
And one auxiliary parameter
```

```
2: Up limit for number of events for each channel.
```

**This script 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 each decay particle and after that on the second cycle generates for each channel the number of events proportion to its partial width. Input parameter for decay case specifies the total number of events for each particle.**

**Generated events are stored in `events_#.txt` files.**

## **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 Showering and Hadronisation Event Generators (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.**

## ***Fusion of collision and decay events in CalcHEP.***

***If a file with decay events of some outgoing particle is detected among ones attached by***

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

***then this decay is implemented passing in decay record information about virtual particle. This procedure works recursively while some decay channel is found.***

***SHG has to know the properties of virtual particles for correct hadronization. They are passed via QNUMBERS (P. Skands et al, hep-ph/0311123) block generated automatically with C-code for all non-SM particles:***

```
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)
```

***There is another way to help SHG to decay BSM particles. It is based of SUSY Les Houches Accord. 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+)
.....
```

***To create such file CalcHEP has a tool for process generation***

`!SM->2*x`

***and a command at numerical session to write or decay modes in Les Houches format.***

***Be continued by A. Belyaev.***

## Code optimization in CalcHEP

*One problem of CalcHEP is a huge size of codes for 2→4 processes. We have realized a option to reduce significantly size of generated code and accelerate numerical calculation.*

*In general CalcHEP compiles code for diagram which mainly is a polynomial over momenta scalar product with coefficients which are functions on model parameters. Now CalcHEP calculates one time for all diagrams all needed products of scalar products and store them in array. Coefficients at this monomial are calculated one time for session and also stored in some array.*

*So, the code for one diagram is extremely short now.*

```
static const int jumps[30]={266,271,272,274,276,285,293,294,393,395,398
399,409,410,418,419,422,424,429,432,433,435,441,442,459,468,477,478,492
};
for(i=0,R=0 ;i<30;i++){R+=CC[i]*DPmonom[jumps[i]];}
```

*Array jumps is used to bypass zero coefficients.*

## **Plans on future.**

*CalcHEP is a program that can be improved in any point. There are two ideas which are related to implementation of decays for CalcHEP event flow.*

*One idea is very primitive, realization of spin correlation. I hope that  $2 \rightarrow 2$  and  $2 \rightarrow 3$  reaction with subsequent  $1 \rightarrow 2$  and  $1 \rightarrow 3$  decays can cover large part of physics. I hope it will work in spite of CalcHEP diagram squaring.*

*The second idea is not so primitive but very easy for realization. I am going to swap the roles of symbolic and numerical modules in CalcHEP. In next version the head module will be numerical one. It will send request for symbolic model to generate libraries for needed matrix elements. These libraries will be linked dynamically and stored on disk for next session. So, to realize decays of outgoing particles in event flow CalcHEP will not need preliminary generated decay events. It will look for needed library. If it is absent, CalcHEP will generate it. If  $1 \rightarrow 2$  decays are absent CalcHEP will generate  $1 \rightarrow 3$  for selected channels ... This scheme efficiently works in micrOMEGAs.*

## **BSM implemented in CalcHEP.**

*During development of micrOMEGAs are realized MSSM(SuSpect, Isajet, SoftSusy, SPheno), NMSSM(NMSSNtools), CPVMSSM(CPSuperH), RHNM*

*The following BSM realizations are known*

*LeptoQuarks(A. Belyaev et al, hep-ph:/0502067)*

*Little Higgs ( A. Belyaev, et al arXiv:hep-ph/0609179)*

*Higgsless 3-site model ( A. Belyaev, S. Chivukula, N. Christensen, E. Simmons et. al. arXiv:hep-ph/arXiv:0708.2588)*

*MUED5, MUED6 ( see Kong WEB page <http://home.fnal.gov/~kckong/mued/>)*

*Almost all these realization were done by means of **LanHEP** package.*