New facilities of CalcHEP_3.6

Alexander Belyaev, Neil D. Christensen, Alexander Pukhov

March 10, 2015

Abstract

We describe new option of CalcHEP package.

1 Default composites and saving of input.

In previous versions of CalcHEP one has to type several times the same setting parameters testing different processes. In the current version we try to simplify user input. First of all entering the processes one can use alias 'p*' for list of partons. For example, if one would like to test production of two neutralinos with a jet in proton-proton collision, the line of processes input could be

```
Enter process:  p*,p* -> ~o1,~o1,p*
```

CalcHEP automatically substitutes `'p*'` contents in the input line

```
composite  'p*' consists of:  G,u,U,d,D,s,S,c,C,b,B
```

At this step the user can edit the `'p*'` contents, it needs, and press the Enter key for confirmation. The names used for gluon and quarks depend are detected automatically according to PDG codes of model particles.

In the same manner at first call of numerical session one gets default table

```
                Composites
 Name  |> Comma separated list of particles
 p*    |G,d,u,U,D,s,S,c,C,b,B
```

and recommended `Cuts`

```
                Cuts
!| Parameter | Min bound   | Max bound
%| T(p*)     | 50          |
%|J(p*,p*)   | 0.5         |
```

These cuts need to remove configurations with small transverse momenta of QCD jets and small angles between them. Without such cuts, in general, one expects infrared divergence of tree-level cross sections. These two cuts are not applied until user removes the commented symbol % in the beginning of the records.

User input for numerical session is saved in file *session.dat* disposed in subdirectory *results*. If the user closes **n_calchep** session and starts it again in some time, the input

is restored. But when the user generates a new process, then directory *results* is cleaned and the setting are lost. In the last version we copy a part of *session.dat* in directory *results/aux*. Then session for new process heritage the model parameters, structure functions and other setting from the previous one. Actually all input is saved except of Vegas parameters. But one has to note that changing of model of particle interaction in symbolic session will lead to lost of user input.

2 Parallel calculations

In the current version we have implemented parallel calculation in both symbolic and numerical CalcHEP sessions. This paralleling is realized in framework of calculation of one process, while paralleling in the *batch* mode is designed for simultaneous calculation of several processes.

Table 1 presents CalcHEP modules where paralleling is implemented and the method of paralleling.

Table 1: Paralleling in CalcHEP

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

Implementing parallel calculations in long-written program we tried to use *threads*, but in some cases it was not possible because of significant use of global variables. In our implementation the daughter processes are not independent. Finishing one step of cycle they ask the parent process which next item should be calculated. We use this way because different steps of cycles need different time for calculation. For instance, a time needed for symbolic calculation of diagram significantly depends on the diagram. Some of Vegas cubes integrated fast because they are excluded by cuts. Thus, it is difficult to share a priory elements of cycles between different daughter processes such that they finish work in the same time. From the other side, we have to note that result of Monte Carlo parallel calculation obtained by this way depends on CPU utilization and, so, is non-reproducible.

We have compared our parallel Vegas with the one included in the CUBA library [1] and based on the *fork* technique. For tested examples CalcHEP Vegas works faster, but the difference is not large. We know that there is a version of Vegas written by Richard Kreckel and based on *pthreads* [2], but we have not compared efficiency of our Vegas implementation with this package.

Initially the number of processors used for calculation is defined by the

```
sysconf(\_SC\_NPROCESSORS\_ONLN)
```

command. It can be changed using menu which can be activated by the **F5** functional

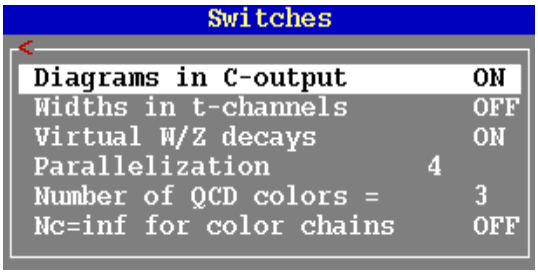


Figure 1: The "Switches" menu in CalcHEP symbolic part

key[1]. See Fig. 1. This menu has a position

```
Parallelization      4
```

which allows one to set the number of processors. The number of used processors in the end of symbolic/numerical sessions is saved in file and restored when CalcHEP is started again. In the beginning of new session CalcHEP also checks environment variable *nParProc* and if such variable exists and contains a number, than this number is used to define number of processors for parallel calculation and has a priority comparing one defined by *sysconf* and stored in files. When symbolic CalcHEP session launches numerical one, it defines environment variable *nParProc* and passes by this way number of processors defined in symbolic session to numerical one.

Because computer resources are restricted, we can have a conflict between paralleling of processes in *batch* and paralleling of calculation in one process. To avoid such conflicts one has to specify in input file of batch session parameters

```
Max number of nodes:              2
Max number of processes per node: 2
```

in a proper way. The first one defines number of different physical reaction calculated in the same time, the second one defines number of processors which will be used for each reactions. The deprecated key `"Max number of cpus"` is still supported but not recommended to use simultaneously with new ones.

Below we present efficiency of paralleling in *GUI* and *batch* calculations.

.....

3 Color module in CalcHEP.

3.1 New color particles and vertices.

We have implemented an option to work with sextet color particles and color antisymmetric **333** vertex. Sextet is defines as a symmetric part of 3×3 tensor product. To implement sextet in CalcHEP one has to type **6** in **cdim**-*color dimension* column of CalcHEP particle table.

CalcHEP substitutes automatically color factor for each vertex depending on color dimensions of particles. The list of supported color vertexes is presented in Table 2. Here

[1]This menu also can be activated by clicking the bottom line of CalcHEP screen. In symbolic/numerical modes it has labels *Switches*/*Options* correspondingly.

we use Latin and Greek indexes for adjoin and fundamental SU(3) representations; $f^{\alpha\beta\gamma}$ presents the structure constants of $SU(3)$ group, $\hat{\tau}^{\alpha}$ - matrices of fundamental representation. They are related by equations

$$\hat{\tau^{\alpha}}\hat{\tau^{\beta}} - \hat{\tau^{\beta}}\hat{\tau^{\alpha}} = i \cdot f^{\alpha\beta\gamma}\hat{\tau}_{\gamma} \quad (1)$$

$$tr(\hat{\tau^{\alpha}}\hat{\tau^{\beta}}) = \frac{1}{2}\delta^{\alpha\beta} \quad (2)$$

Absolutely antisymmetric tensor ϵ is normalized by equation

$$\epsilon^{abc}\bar{\epsilon}_{abc} = 6 \quad (3)$$

Table 2: Color structure of CalcHEP vertexes

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	$(\tau^{\gamma}{}^a_c\delta^b_d + \tau^{\gamma}{}^a_d\delta^b_c + \tau^{\gamma}{}^b_d\delta^a_c + \tau^{\gamma}{}^b_c\delta^a_d)/4$
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	$(\delta^c_a\delta^d_b + \delta^d_a\delta^c_b)/2$

Calculation of color factors in CalcHEP is done for generic $SU(N_c)$ group by technique of color graph reduction. See details in Appendix. By default color factor of squared diagrams is calculated at $N_c = 3$, but also there is an option to select leading N_c^p term, where

$$p = n_8 + \min(n_3 + 2n_6, n_{\bar{3}} + 2n_{\bar{6}}) + \frac{2}{3}|n_3 + 2n_6 - n_{\bar{3}} + 2n_{\bar{6}}| \quad (4)$$

where n_x present numbers of particles corresponding color representation. After that $N_c = 3$ is substituted in the leading term. This option can be activated via F5 functional key, *Switches* menu, line 5. See Fig.1.

To check calculated color weights of diagrams one can activate graphical menu of squared diagrams **"View squared diagrams"**, highlight the needed diagrams and press the **G** key. Then information about different factors involved in calculation of given diagram will be displayed. Color factor is one of them.

3.2 Color chain basis.

For hadronization of generated events one has to describe event color structure. Traditionally the *color chain basis* is used for it. If only 3 and $\bar{3}$ particles are involved in the matrix element and $n_3 = n_{\bar{3}}$, then elements of color basis are constructed by product of Kronecker δ which contract color indexes of 3 and $\bar{3}$ particles. There are $n_3!$ such basis

elements. Diagonal terms of matrix of scalar products of basis elements have order $N_c^{n_3}$. The off-diagonal elements are not zero, but the leading N_c power is smaller for them. Thus, color chain basis is orthogonal only in the $N_c \to \infty$ limit.

Octet can be treated as traceless tensor product of 3 and $\bar{3}$ states. In the color chain basis octet is presented by a $3, \bar{3}$ couple, but the chains (Kronecker δ) which construct 3 and $\bar{3}$ belonging to the same octet are excluded.

Sextet is a symmetric part of tensor product of 2 triplets and in the color chain basis sextet is presented by two triplets. Here one has to take into account that color chains which can be transformed one to other by swapping of triplets belonging to the same sextet, are equivalent. This point significantly decreases a pow of color chain basis for matrix elements with sextets.

Color structure of matrix elements which contain the same number of ϵ and $\bar{\epsilon}$ terms can be described as above because $\epsilon \cdot \bar{\epsilon}$ can be expressed in terms of Kronecker δ. Otherwise the difference between numbers of 3 and $\bar{3}$ indexes proportion to 3 and we involve in color basis the needed number of ϵ^{abc} or $\bar{\epsilon}_{abc}$ terms.

Octets and sextets are naturally can be treated at arbitrary N_c, where they have dimension $N_c^2 - 1$ and $N_c(N_c + 1)/2$ respectively. But ϵ^{abs} is defined only at $N_c = 3$. Its arbitrary dimension extension has N_c indexes and is normalized by condition

$$\epsilon^{abc\ldots}\bar{\epsilon}_{abc\ldots} = N_c! \tag{5}$$

From formal point of view one can define

$$\epsilon^{abc}\bar{\epsilon}_{abc} = N_c(N_c - 1) \tag{6}$$

and observe that this definition leads to orthogonality of color chain basis with ϵ^{abs} terms in $N_c \to \infty$ limit.

Evaluating cross section end generating events CalcHEP does not calculate weights of elements of color basis. But when event is chosen, squared matrix element is recalculating together with color basis coefficients. Then one of basis elements is selected randomly according to the weights. In previously versions of CalcHEP we used $N_c \to \infty$ limit to calculate color weights. In the current version by default $N_c = 3$ if the same option is chosen for matrix elements. To return to old style one has to set

`Nc=inf for color chains` **ON**

in *Switches* menu Fig.1. But if matrix element is calculated in $N_c \to \infty$ limit, the same option will be automatically used for color chains.

Color basis for in CalcHEP C-code output is presented by structure

```
struct  { int pow; int nC; int * chains;} colorBasis;
```

where `pow` describes power of basic, `nC` - number of *chains* in one basis element. Each *chain* in presented by 4 numbers. The first number is 2 for Kronecker δ chains and 3 for ϵ chains. Next numbers present particles included in the chain. The `chains` element of `colorBasis` structure contains $4 \cdot pow \cdot Nc$ numbers which describe color basis. For example, for $G, G \to u, U$ processes CalcHEP writes in results/sqme.c output file

```
static int cwb_1[36]=
      {
       2,1,2,0 , 2,2,1,0 , 2,4,3,0 ,
```

```
        2,1,3,0 , 2,2,1,0 , 2,4,2,0 ,
        2,1,2,0 , 2,2,3,0 , 2,4,1,0
       };
  colorBasis  cb_ext[1]={{ 3, 3,  cwb_1}};
```

One can easily find calculated color weights in C-codes of squared diagrams. For example

```
/*
                G           U       U           G
              -1---\     /==<==   ==<==\     /---1-
                P1 |     |  P4      P4 |     |  P1
                   |     |             |     |
                G  |  G  |  u       u  |  G  |  G
              -2---@-3---@==>==   ==>==@---4-@---2-
                P2    P5    P3      P3   -P6    P2
*/
  ..................................
 if(cb_coeff)
 {
  cb_coeff[1] += (R*4)/(9); /* (1 3) (2 1) (4 2) */
  cb_coeff[2] += (R*4)/(9); /* (1 2) (2 3) (4 1) */
 }
```

Here `cb_coeff` is argument of function

```
  double sqme_ext(int nsub,double GG, REAL * momenta, REAL*cb_coeff,int * err);
```

which returns squared matrix element and `R` presents contribution of given diagram into it. In $N_c \to \infty$ limit sum of coefficients for each diagram should be 1, but for $N_c = 3$ this relation is slightly broken because non-orthogonality of color chain basis.

Implementation of color chains in Les Houches Event (LHE) file is done according to [3, 4, 5]. The uncertainty in treatment of color structures with several ϵ-terms mentioned in [5] is solved by requirement that one has to use set of subsequent numbers to label color tails of one ϵ vertex [6].

4 New functions for hGG and hAA decays

We present new functions to construct effective vertexes for loop induced interaction of Higgs with 2γ and 2 gluons. These functions provide simple implementations of ones presented in our previous paper [7] producing the same results. They are

- `hGGeven(Mh,a, Nloops,...)`
- `hGGodd(Mh, a, Nloops,...)`
- `hAAeven(Mh,a,Nloops,...)`
- `hAAodd(Mh,a,Nloop,...)`

Here Mh is mass of Higgs particle, $a = \alpha_{qcd}(Mh)/\pi$, $Nloops$ - number of loops which contribute to vertex. After the $Nloop$ parameter $4 \cdot Nloops$ parameters are expected. Each 4 parameters determinate contribution of one loop. They should be presented in

the order: `spin2` - spin of particle with factor 2, `cdim` color dimension, `Mp` - pole mass of loop particle, `coeff` - normalized coefficient of vertex of Higgs interaction with loop particle. `coeff` is normalized on mass for fermions, and squared mass for bosons. `hXXeven` functions correspond to scalar interaction, while `hXXodd` ones - to pseudo-scalar one. For example, in SM

$$\lambda_{gg} = hGGeven(Mh, a, 2, 1, 3, Mtp, \frac{1}{v}, 1, 3, Mbp, \frac{1}{v})$$

where v is Higgs vacuum value,`Mtp` and `Mbp` - pole masses of t, and b. Then effective Lagrangian reads

$$L = \lambda_{gg} \cdot hF_{\mu\nu}F^{\mu\nu}$$

Note that parameters specifying the loop do not contain electric charge. Thus squared electric charge has to be included as external factor for $\lambda_{\gamma\gamma}$ effective couplings, for example in SM

$$\begin{aligned} \lambda_{\gamma\gamma} &= hAAeven(Mh, a, 2, 1, 1, Ml, \frac{1}{v}, 2, 1, MW, \frac{-2}{v}) && (7) \\ &+ \frac{4}{9}hAAeven(Mh, a1, 1, 3, Mtp, \frac{1}{v}) + \frac{1}{9}hAAeven(Mh, a, 1, 1, 3, Mbp, \frac{1}{v}) && (8) \end{aligned}$$

The `hAA...` functions contains 1-loop QCD correction as it presented in [7] The `hGG...` functions contains only correction to effective vertex obtained by means of low-energy theorem. Plus to such corrections there are raduative QCD corrections which calculated for point `like hGG` vetrex. In case of `h` decay these correction reads [8]

$$Rqcdh^2 = 1 + a * (149/12 + a * (68.6482 - a * 212.447) \quad (9)$$

This factor `Rqcdh` should be included in vertex together with λ_{gg} to reproduce Higgs width with α_{qcd}^5 precision in tree level $h \to GG$ process.

5 New GUI facilities.

As in previous version font for GUI session is defined by a record in `calchep.ini` file which is started from the keyword **font**. But now it defines not the font itself, but a font family which could be used in GUI session. The font of first window of GUI session is defined by addition record in `calchep.ini` file started from the keyword **width** which allows to extract one font from the selected family. By default `calchep.ini` contains

```
font *normal*
width  10
```

One should use '*' in font specification to select a set which contains more then one element. The user can change size of font by the *Ctrl+* and *Ctrl−* keyboard sequences or *Ctrl>* and *Ctrl<*.

The full list of available fonts can be found by the command `xlsfonts`.

6 Notes about interface with other packages.

Pythia: Pythia8205 reads successfully LHE events generated by CalcHEP. We have tested that Phythia works fine with sextets and **333** color vertices.

LHAPDF: CalcHEP supports interface with LHAPDF-5 and LHAPDF-6 starting from version 6.1.5.

LanHEP: The current version of LanHEP allows to generate model files with sextets and **333** vertices.

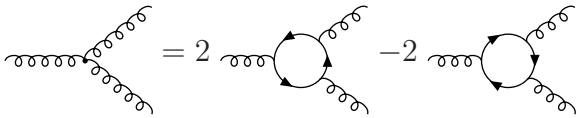
7 Appendix

Here we present rules of color graph technique used for evaluation of color weights.

- Using equation

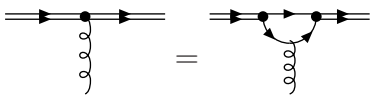
$$-i \cdot f_{abc} = 2 \cdot tr(\hat{\tau}_b \hat{\tau}_a \hat{\tau}_c) - 2 \cdot tr(\hat{\tau}_a \hat{\tau}_b \hat{\tau}_c) \tag{10}$$

  we replace 888 vertex on $8\bar{3}3$ ones:



(10a)

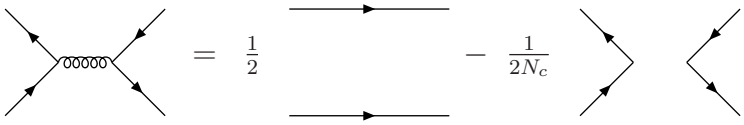
- The $8\bar{6}6$ vertex is replaced on diagram with $6\bar{3}\bar{3}$, $\bar{6}33$ and $8\bar{3}3$:



- The Fiertz identity

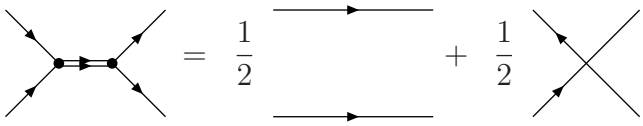
$$(\tau_a)^i_j (\tau_a)^k_l = \frac{1}{2}\delta^i_l \delta^k_j - \frac{1}{2N_c}\delta^i_j \delta^k_l \tag{11}$$

  allows to remove $8\bar{3}3$ vertices:



(11a)

- Sextet line which connected with triplets in both sides is removed by

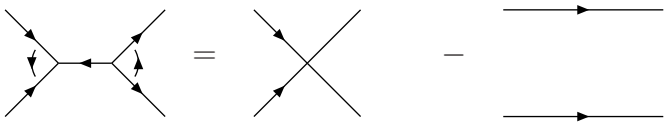


(12)

- Contracted ϵ and $\bar{\epsilon}$ terms remove by equation

$$\epsilon^{abi}\bar{\epsilon}_{icd} = \delta^a_c \delta^b_d - \delta^a_d \delta^b_c \tag{13}$$

that has graphical interpretation



- At final step we replace tadpoles on numbers:

$$= N_c \tag{14}$$

$$= N_c^2 - 1 \tag{15}$$

$$= N_c(N_c+1)/2 \tag{16}$$

References

[1] T. Hahn, Nucl. Instrum. Meth. A **559** (2006) 273 [hep-ph/0509016].

[2] ftp://ftpthep.physik.uni-mainz.de/pub/pvegas/pvegas.c

[3] E. Boos, M. Dobbs, W. Giele, I. Hinchliffe, J. Huston, V. Ilyin, J. Kanzaki and K. Kato *et al.*, hep-ph/0109068.

[4] J. Alwall, A. Ballestrero, P. Bartalini, S. Belov, E. Boos, A. Buckley, J. M. Butterworth and L. Dudko *et al.*, Comput. Phys. Commun. **176** (2007) 300 [hep-ph/0609017].

[5] N. Desai and P. Z. Skands, Eur. Phys. J. C **72** (2012) 2238 [arXiv:1109.5852 [hep-ph]].

[6] Peter Skands, private communication.

[7] A. Belyaev, N. D. Christensen and A. Pukhov, Comput. Phys. Commun. **184** (2013) 1729 [arXiv:1207.6082 [hep-ph]].

[8] P. Baikov, K. Chetyrkin, Higgs Decay into Hadrons to Order alpha**5(s), Phys.Rev.Lett. 97 (2006) 061803.