

# New facilities of CalcHEP\_3.6.28

April 27, 2017

## Abstract

We describe new options of CalcHEP package implemented after CPC publication [1]

## Contents

<b>1</b>	<b>Parallel calculations</b>	<b>2</b>
<b>2</b>	<b>Default composites.</b>	<b>3</b>
<b>3</b>	<b>Saving user setting for numerical sessions.</b>	<b>4</b>
<b>4</b>	<b>New functions for constuction hGG and hAA effective vertices.</b>	<b>4</b>
<b>5</b>	<b>New GUI facilities.</b>	<b>5</b>
5.1	Fonts . . . . .	5
5.2	Mouse . . . . .	5
5.3	Plots . . . . .	5
<b>6</b>	<b>Structure functions</b>	<b>6</b>
6.1	LHAPDF . . . . .	6
6.2	PDT format . . . . .	6
6.3	Options to display and compare structure functions . . . . .	6
6.4	Factorization and renormalization scales. . . . .	6
<b>7</b>	<b>Script subproc_cycle.</b>	<b>7</b>
<b>8</b>	<b>Script par_scan</b>	<b>8</b>
<b>9</b>	<b>Color module in CalcHEP.</b>	<b>9</b>
9.1	New colour particles and vertices. . . . .	9
9.2	Colour chain basis. . . . .	10
<b>10</b>	<b>Appendix</b>	<b>11</b>

# 1 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	fork
Vegas MC integration	threads
Generation of 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 priori 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 [2] 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* [3], but we have not compared efficiency of our Vegas implementation with this package.

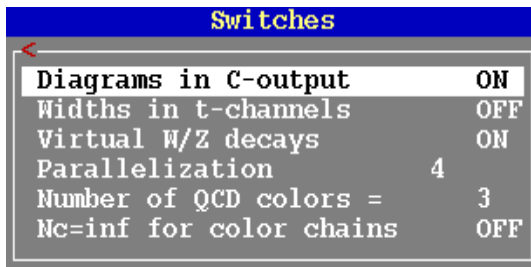


Figure 1: The "Switches" menu in CalcHEP symbolic part

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 key<sup>1</sup>. 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.

.....

## 2 Default composites.

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

---

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

### 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.

## 3 Saving user setting for numerical sessions.

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.

## 4 New functions for constuction hGG and hAA effective vertices.

We present new functions to construct effective vertexes for loop induced interaction of Higgs with  $2\gamma$  and 2 gluons. These functions provide simple implementations of ones presented in our previous paper [1] 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,  $M_{tp}$  and  $M_{bp}$  - pole masses of  $t$ , and  $b$ . Then effective Lagrangian reads

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

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

$$\lambda_{\gamma\gamma} = hAA_{even}(Mh, a, 2, 1, 1, Ml, \frac{1}{v}, 2, 1, MW, \frac{-2}{v}) \quad (1)$$

$$+ \frac{4}{9} hAA_{even}(Mh, a, 1, 1, 3, Mtp, \frac{1}{v}) + \frac{1}{9} hAA_{even}(Mh, a, 1, 1, 3, Mbp, \frac{1}{v}) \quad (2)$$

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

$$R_{qcdh^2} = 1 + a * (149/12 + a * (68.6482 - a * 212.447)) \quad (3)$$

This factor  $R_{qcdh}$  should be included in vertex together with  $\lambda_{gg}$  to reproduce Higgs width with  $\alpha_{qcd}^5$  precision in tree level  $h \rightarrow GG$  process.

## 5 New GUI facilities.

### 5.1 Fonts

Fonts for GUI session are defined in `calchep.ini` file.

```
fontCurrent -adobe-courier-bold-o-normal--18-180-75-75-m-110-iso8859-1
fontFamily *adobe*
fontWidth 15
```

The first line presents the font used in previous session. The second line presents font family used to change font size. The third line contains size of current font. The user can change size of font by the `Ctrl+` and `Ctrl-` keyboard sequences.

One can use "\*" for `fontFamily`, but in this case search of new font can be slow. Full list of fonts available on X-terminal can be displayed on the screen by the `xlsfonts` command.

### 5.2 Mouse

In new version the pressing of mouse button out of menu area is similar to the `Esc` key pressing end leads to exit from menu.

### 5.3 Plots

We have updated routine which present plots in CalcHEP

- There is an option to present several curves on one plot. It is used to compare structure functions presented by various sets.
- There is an option to use logarithmic scale for X-coordinate.
- There are option to export CalcHEP plots to PAW, GNUPLOT and ROOT systems.

## 6 Structure functions

### 6.1 LHAPDF

CalcHEP supports interface with LHAPDF-5 and LHAPDF-6 starting from version 6.1.5. To link with LHAPDF one has to write down path to LHAPDF library `libLHAPDF.so` into environment variable `LHAPDFPATH`. Usually this library is disposed in `Ipath/lib/` where `Ipath` is a place of LHAPDF installation. If LHAPDF is installed in system aria, then `Ipath=/usr`

```
export LHAPDFPATH=Ipath/lib   for sh shell family
setenv LHAPDFPATH Ipath/lib   for csh shell family
```

If `LHAPDFPATH` is properly defined, all installed LHAPDF distributions become available in CalcHEP.

CalcHEP supports photon structure functions included in LHAPDF-6.

### 6.2 PDT format

CalcHEP still supports its own PDT format which is very close to format of LHAPDF-6. CalcHEP has a tool which transforms LHAPDF-6 tables into CalcHEP format. It's `lhpdf2pdt` disposed in `$CALCHEP/bin`. First parameter of this command has to be a name of directory where LHAPDF-6 tables are stored. Usually they are disposed in `Ipath/share/LHAPDF`. For instant

```
$CALCHEP/bin/lhpdf2pdt Ipath/share/LHAPDF/CT10
```

creates file `CT10.pdt` which presents central value for CT10 tables family. One can add one extra parameters to create table for other members of CT10 family. For example,

```
$CALCHEP/bin/lhpdf2pdt Ipath/share/LHAPDF/CT10 5
```

creates file `CT10:5.pdt` which presents set 5 of CT10. Generated PDT files have to be moved to directory `$CALCHEP/pdTables` to be available in CalcHEP sessions. By default this directory contains

```
CT10.pdt           MRST2004qed_proton.pdt
cteq6l1.pdt       NNPDF23_lo_as_0130_qed.pdt
```

All of them were obtained by means of `bf lhpdf2pdt` routine.

### 6.3 Options to display and compare structure functions

New version of CalcHEP provides a possibility to display and compare different structure functions. To get the corresponding plots one has to press `F7` functional key working with "IN state" menu. All possibilities are explained on Fig.2. Actually CalcHEP compares structure functions of two incoming particles.

In the same manner there CalcHEP provides a possibility to compare  $\alpha_{QCD}(Q)$  realization for different PDF sets and compare them with standard  $\overline{MS}$  formula. The corresponding menu and plot are demonstrated on Fig.3

### 6.4 Factorization and renormalization scales.

Menu presented on Fig.3 also provides an opportunity to choose different scales for  $\alpha_{QCD}$  (*renormalization scale*), `pdf1`, and `pdf2` (*factorization scales*). From theoretical point of

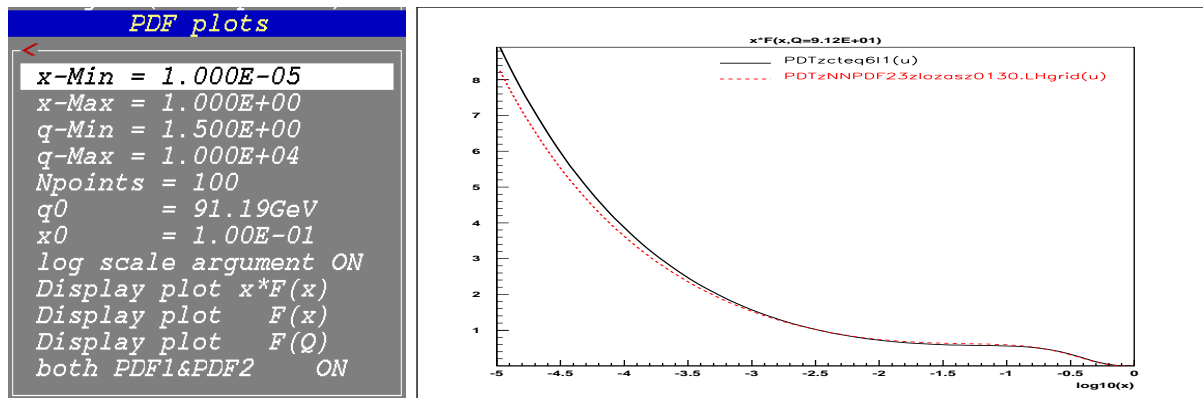


Figure 2: Display and comparison of structure functions.

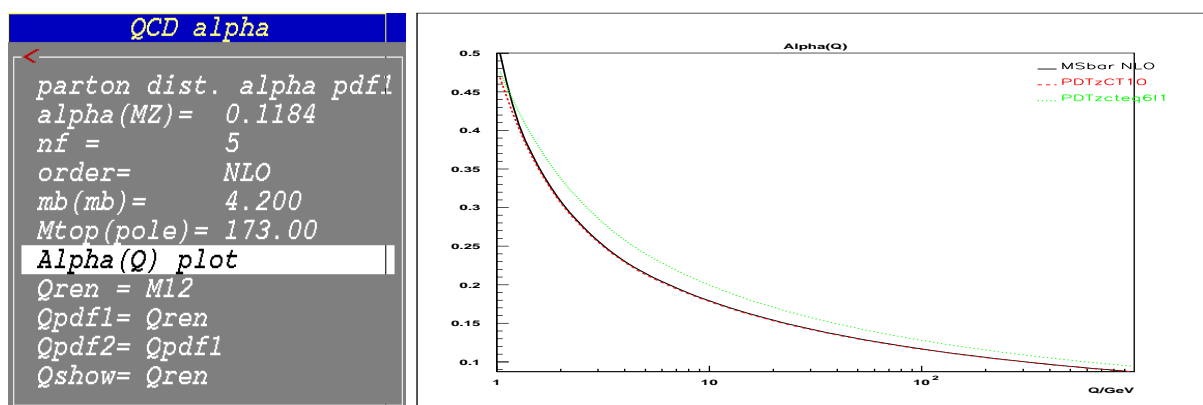


Figure 3: Display and comparison of  $\alpha_{QCD}$ .

view all three scales must match, but sometimes people prefer to distinguish them. In the last case there is a question, what should be the showering scale which is passed to Pythia for hadronization? In current version we present this scale as a free parameter. To simplify typing of four scales mentioned above CalcHEP has an option to define a scale equal one defined above as it is presented on Fig.3.

## 7 Script subproc\_cycle.

This script is significantly improved. Its facilities are not so wide as for `run_batch`, but it is much simpler for use. It is assumed that all settings are defined during GUI session and after that the `subproc_cycle` launched from directory `results` makes a cycle over all subprocesses, calculates cross sections, constructs distributions for various physical variables and performs their summation over subprocess. It also generates events in LHA format and makes a mixture of events generated in different subprocess. For the last case this script has to be called with one addition parameter which defines number of events.

```
subproc_cycle  $N_{events}$ 
```

This command will generate  $N_{events}$  for each subprocess and after that create a mixture file which contains from different subprocesses number of which is proportion to cross section of corresponding subprocess. Other possible call of `subproc_cycle` allows to limit number of generated events by collider luminosity if inverse femtobarn

`subproc_cycle`  $N_{events}$   $L_{fb}$

In this case number of generated events is limited by two parameters. Results produced by `subproc_cycle` session are stored in the following files:

<code>distr_First_Last</code>	contains filled distributions. Here <i>First</i> and <i>Last</i> are numbers of first and last Monte Carlo session launched by <code>subproc_cycle</code>
<code>events_First_Last.lha</code>	contains generated events
directory <code>First_Last</code>	contains <code>prt_#</code> generated during session.
<code>First_Last\run_details.txt</code>	file which contains information about cross sections of all subprocesses.

## 8 Script `par_scan`

This script is interndent for cycle calculation for sets of parameters stored in file. Call of program has to be done from the directory `\results`, where compiled executable file `n_calchep` is disposed. It is assumed that cuts, scale, structure functions, and parameters of integration are specified by the user in GUI mode. The data file should be passed as input

```
$CALCHEP/bin/par_scan < data_file
```

Data file structure should have the following 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`.  $fun\_name_i$  present names of constrained parameters which will be presented in output file . Output file has the same structure as input one but contains calculated numerical values for constrained parameters, an additional column for evaluated cross section with statistical error, and a column which specifies serial numbers of Monte Carlo sessions launched for calculation. Presented cross section corresponds to sum over all subprocesses. If one would not like to control `prt_#` files generated in Monte Carlo sessions, then he can call `par_scan` with additional parameter `clean`:

```
$CALCHEP/bin/par_scan clean < data_file
```

Then the `prt_#` files will be deleted after session. If during Monte Carlo session CalcHEP fills some distributions, then after cycle over subprocesses they are collected in one file `distr_#1-#2` where `#1` and `#2` are serial numbers of first and last sessions for given parametr set.



## 9 Color module in CalcHEP.

### 9.1 New colour particles and vertices.

We have implemented an option to work with sextet color particles and colour antisymmetric **333** vertex. Sextet is defines as a symmetric part of  $3 \times 3$  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 colour dimensions of particles. The list of supported color vertexes is presented in Table 2. Here 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

$$\hat{\tau}^\alpha \hat{\tau}^\beta - \hat{\tau}^\beta \hat{\tau}^\alpha = i \cdot f^{\alpha\beta\gamma} \hat{\tau}^\gamma \quad (4)$$

$$tr(\hat{\tau}^\alpha \hat{\tau}^\beta) = \frac{1}{2} \delta^{\alpha\beta} \quad (5)$$

Absolutely antisymmetric tensor  $\epsilon$  is normalized by equation

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

Table 2: Color structure of CalcHEP vertexes

P1	P2	P3	color structure
$3_a$	$\bar{3}^b$		$\delta_b^a$
$6_{ab}$	$\bar{6}^{cd}$		$(\delta_a^c \delta_b^d + \delta_a^d \delta_b^c)/2$
$8_\alpha$	$8_\beta$		$\delta^{\alpha\beta}$
$3_a$	$3_b$	$3_c$	$\epsilon^{abc}$
$\bar{3}^a$	$\bar{3}^b$	$\bar{3}^c$	$\bar{\epsilon}_{abc}$
$8_\alpha$	$8_\beta$	$8_\gamma$	$-i f^{\alpha\beta\gamma}$
$3_a$	$\bar{3}_b$	$8_\gamma$	$\tau_\gamma^a$
$6_{ab}$	$\bar{6}^{cd}$	$8_\gamma$	$(\tau_\gamma^a \delta_c^b + \tau_\gamma^b \delta_c^a + \tau_\gamma^c \delta_d^b + \tau_\gamma^d \delta_c^b + \tau_\gamma^b \delta_c^a + \tau_\gamma^c \delta_d^a)/2$
$6_{ab}$	$\bar{3}^c$	$\bar{3}^d$	$(\delta_c^a \delta_b^c + \delta_d^a \delta_b^c)/2$
$\bar{6}^{ab}$	$3_c$	$3_d$	$(\delta_a^c \delta_b^d + \delta_a^d \delta_b^c)/2$

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 colour factor of squared diagrams is calculated at  $N_c = 3$ , but also there is an option to select leading  $N_c^p$  term, where

$$p = n_8 + \min(n_3 + 2n_6, n_{\bar{3}} + 2n_{\bar{6}}) + \frac{2}{3} |n_3 + 2n_6 - n_{\bar{3}} + 2n_{\bar{6}}| \quad (7)$$

where  $n_x$  present numbers of particles corresponding colour 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 colour 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.

## 9.2 Colour chain basis.

For hadronization of generated events one has to describe event color structure. Traditionally the *colour 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 colour basis are constructed by product of Kronecker  $\delta$  which contract colour 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  $N_c \rightarrow \infty$  limit.

Octet can be treated as traceless tensor product of 3 and  $\bar{3}$  states. In the colour chain basis octet is presented by a 3,  $\bar{3}$  couple, but the chains (Kronecker  $\delta$ ) 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 colour 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 cardinality of colour chain basis for matrix elements with sextets.

Colour structure of matrix elements which contain the same number of  $\epsilon$  and  $\bar{\epsilon}$  terms can be described as above because  $\epsilon \cdot \bar{\epsilon}$  can be expressed in terms of Kronecker  $\delta$ . Otherwise the difference between numbers of 3 and  $\bar{3}$  indexes proportion to 3 and we involve in colour basis the needed number of  $\epsilon^{abc}$  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  $\epsilon^{abs}$  is defined only at  $N_c = 3$ . Its arbitrary dimension extension has  $N_c$  indexes and is normalized by condition

$$\epsilon^{abc\dots}\bar{\epsilon}_{abc\dots} = N_c! \quad (8)$$

From formal point of view one can define

$$\epsilon^{abc}\bar{\epsilon}_{abc} = N_c(N_c - 1) \quad (9)$$

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

Evaluating cross section end generating events CalcHEP does not calculate weights of elements of colour basis. But when event is chosen, squared matrix element is recalculating together with colour basis coefficients. Then one of basis elements is selected randomly according to the weights. In previously versions of CalcHEP we used  $N_c \rightarrow \infty$  limit to calculate colour 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 colour chains ON**

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

Colour 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  $\delta$  chains and 3 for  $\epsilon$  chains. Next numbers present particles included in the chain. The `chains` element of `colorBasis` structure contains  $4 \cdot pow \cdot N_c$  numbers which describe color basis. For example, for  $G, G \rightarrow u, U$  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 colour 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  $N_c \rightarrow \infty$  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 [4, 5, 6]. The uncertainty in treatment of color structures with several  $\epsilon$ -terms mentioned in [6] is solved by requirement that one has to use set of subsequent numbers to label color tails of one  $\epsilon$  vertex [7].

## 10 Appendix

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

- Using equation

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

we replace 888 vertex on 833 ones:

$$= 2 \text{ (diagram 1)} - 2 \text{ (diagram 2)} \quad (10a)$$

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

$$\text{Diagram} = 2 \times \text{Diagram}$$

- The Fiertz identity

$$(\tau_a)_j^i (\tau_a)_l^k = \frac{1}{2} \delta_l^i \delta_j^k - \frac{1}{2N_c} \delta_j^i \delta_l^k \quad (11)$$

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

$$\text{Diagram} = \frac{1}{2} \text{Diagram} - \frac{1}{2N_c} \text{Diagram} \quad (11a)$$

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

$$\text{Diagram} = \frac{1}{2} \text{Diagram} + \frac{1}{2} \text{Diagram} \quad (12)$$

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

$$\epsilon^{abi} \bar{\epsilon}_{icd} = \delta_c^a \delta_d^b - \delta_d^a \delta_c^b \quad (13)$$

that has graphical interpretation

$$\text{Diagram} = \text{Diagram} - \text{Diagram}$$

- At final step we replace tadpoles on numbers:

$$\text{Diagram} = N_c \quad (14)$$

$$\text{Diagram} = N_c^2 - 1 \quad (15)$$

$$\text{Diagram} = N_c(N_c + 1)/2 \quad (16)$$

## References

- [1] A. Belyaev, N. D. Christensen and A. Pukhov, *Comput. Phys. Commun.* **184** (2013) 1729 [arXiv:1207.6082 [hep-ph]].
- [2] T. Hahn, *Nucl. Instrum. Meth. A* **559** (2006) 273 [hep-ph/0509016].
- [3] <ftp://ftphhep.physik.uni-mainz.de/pub/pvegas/pvegas.c>
- [4] E. Boos, M. Dobbs, W. Giele, I. Hinchliffe, J. Huston, V. Ilyin, J. Kanzaki and K. Kato *et al.*, hep-ph/0109068.
- [5] 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].
- [6] N. Desai and P. Z. Skands, *Eur. Phys. J. C* **72** (2012) 2238 [arXiv:1109.5852 [hep-ph]].
- [7] Peter Skands, private communication.
- [8] P. Baikov, K. Chetyrkin, Higgs Decay into Hadrons to Order  $\alpha^5(s)$ , *Phys.Rev.Lett.* 97 (2006) 061803.