

Community contribution

This page contains CompHEP community contributions. Please contact us (comphep@theory.sinp.msu.ru) if you wish to put some extra modules for CompHEP on the site.

Please sign your contribution and indicate a contact information.

Program to plot CompHEP-distributions in ROOT

- imposes two or more curves on the same plot
- draws normalized distributions
- uses LATEX notations
- download: [tab_plots.tgz](#)

— Maxim Perfilov 21/01/2007 05:35

CERN LCG MCDB Project

If you generate event samples for LHC or you are looking for some samples, probably, you will be interesting in [CERN LCG MCDB project](#)

Short Abstract about LCG MCDB: The LCG MCDB project is developed to store and document sophisticated event samples simulated for the LHC collaborations by experts. In many cases, the modern Monte-Carlo simulation of physics processes require expert knowledge in Monte-Carlo generators or significant amount of CPU time to produce an event sample. MCDB is a knowledge base mainly dedicated to accumulate such type of simulated events. The main motivation behind the LCG MCDB project is to make sophisticated MC event samples and their structured description available for various physics groups. All the data from MCDB is accessible in several convenient ways.

— Lev Dudko 25/02/2007 05:35

Script to make Root ntuple for CompHep events

The script **comphep_events2ntuple.pl** allows to create Root ntuple for events, generated by CompHep, and coming in PEV type file, which name is a script argument.

Example of usage :

```
comphep_events2ntuple.pl Mixed.PEV
```

On output it gives two files :

```
comphep.dat      -- data for ntuple
doCompHepNTuple.C -- macro to be run in Root
```

File sizes for 11.5K ttbar events at LHC :

```
comphep.dat      - 5.6M
CompHepData.root - 1.6M
```

To create ntuple run the macro in Root :

```
.x doCompHepNTuple.C
```

Documentation. Script is selfdocumented. To get help, run it with '-h' option :

```
comphep_events2ntuple.pl -h
```

Created ntuple file **CompHepData.root** contains tree **CompHepNtuple** with **following structure**

[var_name (type) - what] :

- IP (I) - process ID
- CS (F) - process CrosSec
- Qsq (F) - Qsquared
- CStot (F) - total CrosSec

—beams:

- bm1KF (I) - beam-1 KF
- bm1M (F) - beam-1 Mass
- bm1E (F) - beam-1 E
- bm1PDF (C) - beam-1 PDF (combined)
- bm1PDFfile (C) - beam-1 PDFfile
- bm2KF (I) - beam-2 KF
- bm2M (F) - beam-2 Mass
- bm2E (F) - beam-2 E
- bm2PDF (C) - beam-2 PDF (combined)
- bm2PDFfile (C) - beam-2 PDFfile

—partons:

- Np (I) - process Nparton
- KF1 (I) - KF for initial parton-1
- m1 (F) - Mass for initial parton-1
- pz1 (F) - Pz for initial parton-1
- KF2 (I) - KF for initial parton-2
- m2 (F) - Mass for initial parton-2
- pz2 (F) - Pz for initial parton-2

—For every outgoing parton / particle (3 - for the first) :

- KF3 (I) - KF
- m3 (F) - Mass
- px3 (F) - Px
- py3 (F) - Py
- pz3 (F) - Pz
- pt3 (F) - Pt

- e3 (F) – E
- phi3 (F) – Theta
- tet3 (F) – Phi
- y3 (F) – Y
- eta3 (F) – Eta

download: comphep_events2ntuple.pl.gz (Version date: Fri Jun 5 15:28:33 MSD 2009)

Enjoy! — Vyacheslav Rud 10/04/2008 19:16

Script to convert OLD pev format into NEW one

The script **convert_comphep_old2new.pl** allows to convert CompHep OLD format (cpyth-1.2.x) .pev file into NEW one (cpyth-2.0.x). Rarely, but sometime you need that. Obvious example is a needness to process CompHep generated files, stored in above mentioned CERN LCG MCDB in **old** format, in a framework, wich requires input CompHep events in **new** format.

The script accepts one argument (old.pev file) and writes out .pev file in new format (new.pev file) :

```
./convert_comphep_old2new.pl tq_tqb_lhc_e.pev
```

Output file tq_tqb_lhc_e.new.pev will be produced.

Documentation. Script is selfdocumented. To get help, run it with '-h' option :

```
convert_comphep_old2new.pl -h
```

Not all variables used in new format available in old one. So, they were predefined to some default values. Correcr them if they don't match your needs.

download: convert_comphep_old2new.pl.gz

— Vyacheslav Rud 05/05/2008 16:07

Script for including cuts, kinematics, or regularizations into batch.dat file

When one need to calculate in CompHep complecate process with many subprocesses, he should define in CompHep GUI cuts, kinematics, and regularizations for every subprocess. These definitions, saved in results/session.dat for selected subprocess, on next step could be shared with other similar subprocesses running num_batch.pl script with -proc option. And that procedure (GUI - num_batch.pl calls) one should repeate many times, manually changing cuts, kinematics, or regularizations, because even when number of unique cuts, kinematics, and regularizations is small, they distributed between subprocesses differently and number of such combinations (and GUI - num_batch.pl calls) is not small. As a consequence, that is a long and tedious work.

That motivated me to write script **tune_batch_dat.pl**, which takes unique information blocks from external file and inserts them into results/batch.dat file into subprocesses, listed for each block. This

external file preparation is easy - blocks format one takes from batch.dat file, cuts, kinematics, and regularizations for every subprocess you know anyway from physics.

This script could be used only after at least one CompHep GUI run with results/batch.dat file creating.

Short description.

Arguments :

```
1 -- batch.dat location
2 -- file with corrections
```

Example : `./tune_batch_dat.pl results/batch.dat batch.dat.tqb.corrections`

Output files :

```
results/batch.dat      -- corrected file
results/batch.dat.orig -- saved original file
```

Documentation. Script is selfdocumented. To get help, run it with '-h' option :

```
./tune_batch_dat.pl -h
```

Along with some explanations this help shows sample **batch.dat.tqb.corrections** file.

Lines like

```
Cuts: 17-32
Kine: 1-2,7-8,11-14,17-18,23-24,27-30
Regs: 17-18,23-24,27-30
```

precede each block and contain obvious key words and list of subprocesses to which the block need to be assigned. Blocks include terminating lines (empty line for Kine, lines with equal signes for Cuts and Regs). Also used special terminators like EOKine, EOParms.

If you would like to replace for all subprocesses some parameters, which stored in batch.dat file in lines of following structure

<key> <value>

you may use Parms block (with any number of lines) :

Parms:

```
SQRT(S) 1.000000E+04
```

EOParms

This sample block sets process sqs energy to 10000 GeV. SQRT(S) will be considered as a key. Of course, one such replacement more easely could be done in text editor!

Obviously, that script works instantly.

Download: tune_batch_dat.pl.gz (Version date: Jun 30 12:42:18 MSD 2008)

— Vyacheslav Rud 01/07/2008 15:35

Script for CompHep cross-sections energy dependence calculation

When you done CompHep calculations for the process under study at one energy, you may easily get cross-sections energy dependence running script **cs_energy_dep.pl**. The script also can produce plots for relative cross-sections for subprocesses.

Short description.

This script could be used only with fully prepared by CompHep results/batch.dat file.

For every energy point the script modifies results/batch.dat file and runs CompHep script **num_batch.pl** to calculate given process cross-sections. On output it creates Root ntuple and macros to produce appropriate cross-section plots.

Arguments :

```
1 -- min energy in CMS (in GeV)
2 -- max energy in CMS (in GeV)
3 -- step in energy      (in GeV)
4 -- process type       (string)
```

Example : `./cs_energy_dep.pl 10000 14000 4000 tqb`

Output files :

```
results_vs_sqs/      -- working directory (copy of results/)
cs_vs_sqs.dat        -- input file for ntuple creation
doCS_vs_sqsNTuple.C -- Root macro for ntuple creation
cs_vs_sqs_plots.C   -- Root macro for CS-s plotting
cs_vs_sqs_plots.ps  -- final CS plots
```

There is important for calculation accuracy parameter `$slope` which defines number of Vegas calls at every energy point - tune it for your needs (the higher `Ecms` the more dense grid needed for numerical calculations).

The macro `cs_vs_sqs_plots.C` has internal switches `ifTeV`, `ifRelCS`. To get relative cross-sections set `ifRelCS` to true.

Script is selfdocumented. To get help, run it with '-h' option :

```
./cs_energy_dep.pl -h
```

Enjoy!

Download: cs_energy_dep.pl.gz (Version date: Jul 8 15:07:45 MSD 2008)

Sample plots: [cs_vs_sqz_plots_log_h1.ps.gz](#), [cs_vs_sqz_plots_std_h2.ps.gz](#)

— Vyacheslav Rud 01/07/2008 15:55

From:

<https://theory.npi.msu.su/> - **THEORY**

Permanent link:

<https://theory.npi.msu.su/doku.php/comphep/extra/main?rev=1483272088>

Last update: **01/01/2017 15:01**

