QCD and Monte Carlo Generators

H. Jung

Lecture course
University Antwerp

ROOT in 5 seconds

- To actually run ROOT, just type:
  `root`
- The shell in which you type this command will become your ROOT command window. First you'll see the white-and-blue ROOT window appear on your screen. It will then disappear, and a brief "Welcome to ROOT" display will be written on your command window.
  If you grow tired of the introductory graphics window, type "`root -l`" instead of "`root`" to start the program.

  You can type "?" (or ".h") to see a list of ROOT commands... but you'll probably get more information than you can use right now. Try it and see.

- quit the ROOT session at the root prompt line with:
  `.q`

- At root [0] prompt line type:
  `TBrowser b;`

  This creates a TBrowser object, and it will appear in a separate window on your screen. The browser lists everything that is part of the current ROOT session.

- Open the root file.
  To open the root file from the Browser, select the File menu and the Open menu item.
  Click on the file `output.root` and click on the Open button
  Double click on the histo you want to view.

- to print the histo into a eps or ps file, go to the display window and click on
  `Save as` and then select the file format you want

Howto book, fill and normalize histograms

- *sketch of a small program:*

```
// include the header files
#include "TH1.h"
#include "TFile.h"
// booking a 1-dim histogram:
   TH1F *h2 = new TH1F("pt","pt",100, 0., 200.);
// event loop start
// fill histogram with pt from HEPMC record:
   h2->Fill( (*p)->momentum().perp() );
// end of event loop
// write histogramm out to file
   TFile f("output.root","RECREATE");
   h2->Write();
   f.Close();
```

- *adding two histograms:*
  assume we have histo h1 and histo h2 and want to add them according to h3 = a*h1 + b*h2

```
TH1F *h3 = new TH1F(*h2);
h3->Add(h1,h2,a,b);
```

- *normalising a histogram to the total cross section:*
  assume we have `NEV` number of total generated events, which corresponds to a cross-section `xsec`. Then normalising a histogram `h1` to the total cross-section can be done with:

```
      TH1F *h2 = new TH1F(*h1);
    a=xsec/NEV;
    b=0;
    h2->Add(h1,h1,a,b);
```

- *normalising a histogram to equidistant bin width:*
  as above, but now with:

```
     a=xsec/NEV/binwidth;
```

- *fitting a histogram with any function:*
  click (right mouse) on the histo in the browser window, select FitPanel*, select type:* Predef-1D*, click on fit.*
  To display fit parameters, click in the display canvas on option and tick mark FitResults

- **Introductory tutorials and further infos:**
  a nice introductory tutorial can be found under:
  http://www-ekp.physik.uni-karlsruhe.de/~vest/teachings/RootTutorial.html

  more advanced ROOT tutorials are under:
  http://root.cern.ch/root/Tutorials.html
  http://www.nevis.columbia.edu/~seligman/root-class/

- For p[eople knowing PAW here is simple description of commands:
  http://root.cern.ch/root/HowtoConvertFromPAW.html

- If you want to know some details on the libraries ROOT uses try:

```
root-config --cflags
root-config --libs
```