PAW

– Introduction to the PAW software package.

Physics Analysis Workstation (PAW) is a piece of software developed at CERN for fitting and visualization of data. It is neither the newest nor the smartest, but it has an easy interface and it is relatively simple to use.

In PAW you type as much of a command as is neccessary for the command not to be misunderstood. `histogram/plot` for instance becomes `h/pl` and `vector/create` becomes `v/cre`. By typing `help` you get a list of the commands and their effect. In PAW as well as in this document braces "[ ]" indicate that something is optional. Options and settings are controlled with the command `set` and `opt`.

h/file LUN filename `h/file 1 distributions.hbook`
This command is used to open a file, *filename*, that contains histograms. The Logical Unit Number (LUN) can be chosen freely and is used later in commands regarding the file (ex.: *close LUN*). However the histograms are also written to memory. To erase them write: `h/del 0`. It is here appropriate to mention that PAW does *not* understand capital letters, so use only lower-case letters.

h/pl ID [c,e,s,lego,surf] `h/pl 1 e`
Plots the histogram ID. The options are: *c* continuous – as a curve, *e* errors – with error bars, *s* superimpose – plot on top of the previous histogram, *lego* lego – a lego-style plot, *surf* surface – surface plot. The *lego* and *surf* options are used with 2D histograms.

h/fit ID[interval] function `h/fit 1(5:15) g`
Fits the histogram ID. PAW has the following built in functions: *g* Gaussian, *e* exponential, *pX* polynomium of order X (ex.: *p3*). Other functions are fitted as: *h/fit ID function ! Nparameters parameter-vector*, where *function* is a user defined function in Fortran. `h/fit 1 func.f !  3 par`.

v/cre name(Nparameters) [R values] `v/cre par(3) R 100.0 5.0 0.1`
Defines a vector, /emphpar with three entries. The optional *R* gives the possibility to enter values directly. Arrays are defined in the same way; `v/cre data(3,10)`.

v/read vector filename `v/read par data.txt`
Loads a file '*filename*' into a vector '*vector*'. If the vector does not exist it will be defined.

v/fit x y ey function `v/fit xvalue yvalue yerror p3`
Fit points with x-values, *x*, y-values, *y* and an error on the y-values *ey* with the function, *function*. Unfortunately the function is not very graphically satisfactory and you will often have to visualize your data in another way.

zone [x,y] `zone 2 3`
Splits the plot window in subwindows. Given without an argument the command restores the original setting (1,1).

opt stat and *set stat* `set stat 0010`
opt stat shows the statistics of the histogram in the top right corner of the plot, and *set stat* specifies which information is to be shown. In the example only the number of entries in the histogram are shown.

opt fit and *set fit* `set fit 1111`
opt fit Shows the parameters of the fit in the top right corner of the plot, and *set fit* specifies which information is to be shown. In the example all parameters are shown with errors.

for/file LUN filename and *meta LUN* `for/file 99 plot.eps` and `meta 99 -113`
Saves the plot to the file *filename*. The command must be followed by *close LUN*. These commands gives the opportunity to save visualized data to a file.

exe filename.kumac `exe fit.kumac`
Executes the commands in the file *filename*. By common occurences of series of commands this can save time. Especially when building complicated plots.

shell UNIX command `shell make`
This command executes a UNIX-command without exiting PAW.

Example of a sequence that fits and plots two histograms in the file ***`data.hbook`***:

`h/file 1 data.hbook`	Opens the file
`opt fit`	Prints parameters from the fit in the plot.
`set fit 1111`	Prints parameters from the fit with errors.
`zone 1 2`	Splits the plot window in two.
`h/fit 10 g`	Fits the histogram ID=10 with a Gaussian.
`h/pl 20 lego`	Plots the histogram med ID=20 “lego-style”.

Example of a sequence that fits and plots a vector in the file ***`values.txt`***:

`v/cre data(3,20)`	Defines the vector `data`.
`v/read data values.txt`	Loads data into the vevtor `data`.
`v/copy data(1,1:20) x`	Copies the x-values from `data` to a new vector, `x`.
`v/copy data(2,1:20) y`	Copies the y-values from `data` to a new vector, `y`.
`v/copy data(3,1:20) ey`	Copies the y-errors from `data` to a new vector, `ey`.
`v/fit x y ey p1`	Fits the points with a linear function.