Fermi National Accelerator Laboratory

FERMILAB-Conf-89/143

# Physics Analysis Workstation *

Harald Johnstad
Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510 U.S.A.

June 1989

* Presented at "Real Time Computer Applications in Nuclear, Particle, and Plasma Physics," Williamsburg, Virginia, May 16-19, 1989.

Operated by Universities Research Association, Inc., under contract with the United States Department of Energy

# Physics Analysis Workstation

Harald Johnstad
Fermi National Accelerator Laboratory[1]
Batavia, Illinois 60510.

Abstract

The Physics Analysis Workstation (PAW) is a high-level program providing data presentation and statistical or mathematical analysis. PAW has been developed at CERN as an instrument to assist physicists in the analysis and presentation of their data. The program is interfaced to a high-level graphics package, based on basic underlying graphics. 3-D graphics capabilities are being implemented. The major objects in PAW are 1 or 2 dimensional binned event data with fixed number of entries per event, vectors, functions, graphics pictures, and macros. Command input is handled by an integrated user interface package, which allows for a variety of choices for input, either with typed commands, or in a tree structure menu driven mode.

## PAW

The Physics Analysis Workstation (PAW) [1] is a modular interactive system of various programming tools and subroutine packages optimized to assist the physicist programmer in the analysis and presentation of data. PAW is written in FORTRAN and has been implemented on most computer systems used in the High Energy Physics community. Although PAW may be used on a conventional terminal, the full functionality of PAW can only be explored in a workstation environment.

### Graphical Functions

The graphical functions in PAW are based on a high-level interface to underlying basic graphics systems, such as the ANSI X3.124-1985 [2] and ISO 7942-1985 [3] Graphical Kernel System (GKS) standards, and the CORE graphics system [4]. The high-level graphics interface allows for dynamic creation of data structures at execution time, as well as manipulation of those structures, and their transport to and from external media on the same or different computers, memory to memory, memory to/from disk, or over the network.

[1]Fermilab is operated by Universities Research Association, Inc., under contract with the U.S. Department of Energy

There is a built-in interface to the PostScript graphics language from Adobe Systems, Inc. [5], which allows for the detailed description of page layouts. PostScript is built into the hardware of a range of printers/plotters, such as the Digital ScriptPrinter and Apple LaserWriter.

High-level graphics macroprimitives, and graphics editing are built-in features of the high-level graphics interface.

### User Interface

The dialogue between the user and the application program is handled by a user interface package, allowing for different styles of dialogue, command mode or a tree structure menu driven mode. Commonly used commands may be referenced via aliases or in a graphics panel. Control statements may be entered via macros, containing a sequence of PAW commands to be executed. Parameter values may be passed to a macro.

### Histogramming Facility

The histogramming facilities in PAW include one- and two-dimensional histograms and Ntuples. An Ntuple is a two-dimenisonal array characterized by a fixed number (N), specifying the number of entries per element, and by a length, giving the total number of elements. Other histogramming features include projections and slices, operations and comparison of histograms, minimization and parametrization tools, random number generation, histogram data base on direct access files, graphics options such as contour histograms, shaded histograms, error bars, smoothed curves and surfaces, scatter plots, surface plots, windowing, and graphics input.

Mathematics

The System for Interactive Graphical Mathematical Applications (SIGMA) [6] array manipulation language is integrated into PAW. In SIGMA, the basic data units are scalars and one or more dimensional rectangular arrays, which are automatically handled. The computational operators resemble those for numerical mathematics. The procedural operators are like those of FORTRAN, providing graphical display of arrays as a set of curves.

FORTRAN Interpreter

PAW uses a FORTRAN interpreter, which allows the user to execute interactively a set of FORTRAN routines in interpretive mode. The interpreter implements a large subset of the complete FORTRAN language and allows for FORTRAN call statements to most subroutines included in the PAW system. With the FORTRAN interpreter, the user may specify selection criteria or a minimization function to be applied to the data analysis.

Remote Login

PAW allows for remote login and remote shell commands and relies currently on the TCP/IP protocol (Transmission Control Protocol / Internet Protocol) which provides remote login (TELNET) and file transfer (FTP). Computers on the network are defined by their network address. TCP/IP will be replaced by the Open System Interface (OSI) standard protocols in the future. This will allow the user to fully utilize the capabilities of a local workstation, while experimental data are being created, stored, and transported from a larger mainframe.

X Windows

An X Windows user environment will be integrated into PAW. The X Windows system is based on a client-server model; multiple clients can have simultaneous connections to a given server. The X Windows system needs to be fully integrated into the lower layers of the operating system kernel to boost its perfomance. As X Windows becomes faster and at least comparable to using the native window manager on a given hardware system, it will become an important enhancement to PAW environments.

## PAW Demo

A PAW demo was presented at the IEEE RT89 conference, using a VAXstation 3200 on a color monitor, running VAX/VMS 5.1, with VWS 4.0 windowing system and DEC GKS V4.0. Due to technical limitations, the intended display of commands could not be made sufficiently legible on the overhead screen. The commands which were used during the demo are included in the following with several in-line comments.

The following corresponding data files which are being referenced in this demo, may be copied from the author upon request (BITnet address: Johnstad@FNAL):

HEXAM.DAT - histogram and Ntuple data
GRADES.DAT - Ntuple data
VECTOR.DAT - vector with 3x10000 elements
TITLE.KUMAC - macro to display title
HPLOT2.KUMAC - superimposed histograms
PANEL.KUMAC - macro to invoke panel mode

The following COMIS (FORTRAN interpreter) routine must then also reside on a local disk with file name ADDN.FOR in the local user area:

```
SUBROUTINE ADDN(N)
SUM=0.
DO I=1,N
   SUM=SUM+I
ENDDO
TYPE SUM
END
```

A pointing device is used in this demo to select menu options and define graphical cuts; a three-button device (mouse) was used during the presentation. The leftmost button on this device is the SELECT button, referenced as MB1 in the following. The middle button, referenced as MB2 in the following, is used to end a graphics input session and return the program to command mode input.

The following is a complete list of PAW commands used in the IEEE RT89 demo presentation. Text strings following a vertical line character are in-line comments.

PAW | Start PAW session
41 | workstation id VAXstation II
exe title WELCOME | init welcome
exe title HELP | init help
help | get help
q | return to command mode
help usage | help on usage
help manual | help on manual
/ | list all commands
a | show ambigiuties
help hist/plot | help on plot command
hist/file 3 hexam.dat | open data file
set ymgu 5
hist/plot 110 | plot id=110
set hcol 1002 | set color to red
hist/plot 110 | plot id=110 in color
exe title MENUS | init menus
style gw | menu driven mode
MB1 histogram | click on histogram menu
MB1 plot | click on plot menu
MB1 id | click on histogram id
110 | enter id=110
MB1 execute | plot histogram id=110
MB2 | terminate graphics input
exe title PANEL | init panel mode
exe panel | input panel
MB1 h/pl | click on h/pl in panel
MB1 h/pl | click on h/pl in panel
MB2 | terminate panel mode
help style | list input modes
exe title MATH | init mathematics
opt grid | set grid option on
cd //pawc | current directory is memory
fun/plot sin(x)/x 0 10 | function plot
exe title VECTORS | init vectors
ve/cre x(10) | create vector with 10 elements
ve/inp x 1 2 3 4 5 5 4 3 2 1 | input 10 elements
zone 1 2 | turn on windowing
ve/dr x 1 | draw vector, copy to histogram id=1
fit/gauss 1 | fit histogram id=1 with a gaussian
hi/pl | plot current histogram id=1
zone | reset windowing
ve/del * | delete all vectors
ve/read x,y,z vector.dat | input vector from file
ve/pl x | plot element x
ve/pl z | plot element z
set htyp -1 | set histogram fill area
ve/pl z | plot element z again
set hcol 1002 | set histogram color red w/fill
sigma w=sqrt(z) | invokes sigma
ve/pl w 2 s | superimpose vector w
exe title COMIS | init comis
shell type addn.for | type comis function
call addn.for(100) | call comis function
call addn.for(1000) | call comis function again
exe title HPLOT | init graphics
cd //lun3 | set cdir to file HEXAM.DAT
hi/pl 110 | plot id=110
call hprint(110) | print id=110
surf 20 29 67 | surface plot id=20
contour 20 25 0 | contour plot id=20
cd | reset current directory
ldir | list current directory
cd ntuples | set current directory to ntuples
opt stat | invoke real-time statistics option
nt/plot 30.x ! ! ! 10 | plot x in ntuple id=30
opt nsta | turn off real-time statistics option
exe title MACROS | init macros
hi/del 0 | delete all histograms
exe hplot2 | execute macro hplot2.kumac
set * | reset graphics options to default
opt * | reset HPLOT options to default
exe title NTUPLES | init ntuples
set ymgu 5 | set upper Y margin
hist/file 1 grades.dat | open ntuple data file
ldir | list current directory
nt/print 60 | print summary of ntuple id=60
set hcol 1002 | set histogram color to red

```
set htyp -1 | set histogram fill area type
set vfon -92 | set font and precision
set vsiz 0.4 | set axis values size
nt/pl 60.netsal | plot 'netsal' in ntuple id=60
nt/pl 60.netsal/1.67 | scale with fact=1.67
nt/pl 60.age | plot 'age' in ntuple id=60
smk 3 | set marker type
nt/pl 60.netsal%age | plot 'netsal' vs 'age'
spmci 2 | set polymarker color index
cut 1 g | define graphics cut
MB1 | define cut border (repeat)
MB2 | terminate grahics cut definition
opt stat | statistics option on
nt/pl 60.netsal 1 | plot 'netsal' for cut=1
nt/pl 60.netsal .not.1 | plot residual data
nt/pl 60.netsal age<40 | define new cut and plot
nt/pl 60.netsal age>40 | define new cut and plot
nt/pl 60.netsal age<40.and.1 | combine cuts
nt/pl 60.netsal age>40.and.1 | combine cuts
nt/pl 60.netsal | plot variable 'netsal' again
set hcol 1002 | set histogram color to red
nt/pl 60.netsal grade<10 ! ! ! s | superimpose
set hcol 1003 | change color to green
nt/pl 60.netsal grade=12.and.children=2 ! ! ! s
exe title END | init end
exe title REPLAY | init replay
set * | reset graphics options to default
opt * | reset HPLOT options to default
ve/del * | delete all vectors from memory
units | check all logical units in use
close 1 | close logical unit 1
close 3 | close logical unit 3
hi/del 0 | delete all histograms from memory
cd //pawc | set current directory to memory
last 200 rt89 | record lass 200 commands
```


## Acknowledgements

It is a pleasure to thank R.Brun at CERN for making PAW available to the High Energy Physics Community, and for his help and comments in preparation of this presentation at IEEE RT89.

It is also a pleasure to thank L.Roberts for invaluable help during implementation of this program at Fermilab, and for help in the preparation of this article.


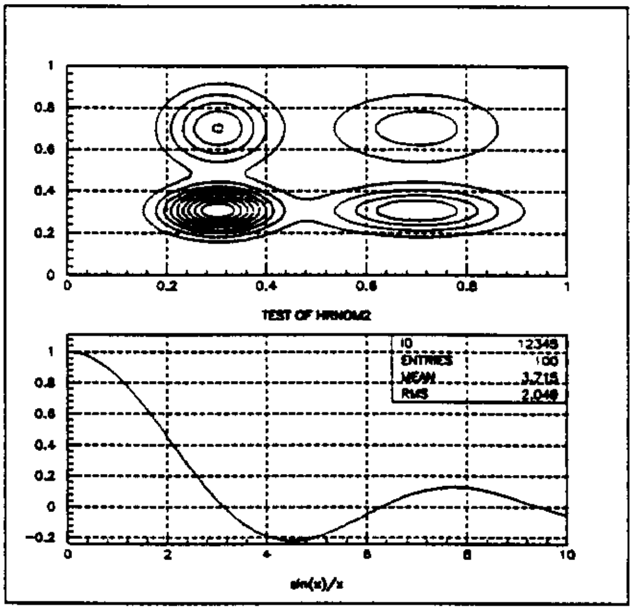


Figure 1: Example of PAW contour plot and function plot.

## References


[1] PAW - Physics Analysis Workstation, CERN Computer Centre Program Library Long Write-Up, R.Brun, O.Couet, N.Cremel, C.Vandoni, P. Zanarini, (September 1988).

[2] American National Standard Institute, American National Standard for the Graphical Kernel System (GKS), ANSI X3.124-1985.

[3] Information Processing - Graphical Kernel System (GKS), Functional Description, International Standards Organization, 7942-1985.

[4] The CORE Graphics System, SIGGRAPH-ACM Computer Graphics 13, 3, (August 1979).

[5] Adobe Systems Incorporated, PostScript Language, Addison-Wesley, 1985.

[6] R.Hagedorn and J.Reinfelds, SIGMA without effort, CERN 78-08, Data Handling Division and Theoretical Physics Division.