Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

POW: A Tcl/Tk Plotting and Image Display Interface Tool for GUIs

L. E. Brown

Hughes STX for NASA/GSFC, Code 664, Greenbelt, MD 20771, E-mail: elwin@redshift.gsfc.nasa.gov

L. Angelini

USRA for NASA/GSFC, Code 664, Greenbelt, MD 20771, E-mail: angelini@lheavx.gsfc.nasa.gov

Abstract. We present a new Tcl/Tk based GUI interface tool which features plotting of curve and image data and allows for user input via return of regions or specific cursor positions. The package is accessible from C, Tcl, or FORTRAN. POW operates on data arrays, passed to it as pointers. Each data array sent to POW is treated as either an Image object or a Vector object. Vectors are combined to form Curves. Curves and Images may then be combined to form a displayed Graph. Several Graphs can be displayed in a single Tk toplevel window. The Graphs can be rearranged, magnified, and zoomed to regions of interest by the user. Individual graph axes can be "linked" to implement a "multiple y-axis" (or x-axis) plot. The POW display can be written out in PostScript, for printing.

1. Introduction

With the development of several GUI based tools for High Energy Astrophysics at the HEASARC, we found a need for a interface tool that supports combined curve plotting and image display in a native Tcl/Tk environment. POW is our answer.

2. Design Goals

The principal design goals for POW were:

1. must coexist with (almost) any language (FORTRAN, C, Tcl/Tk),
2. must allow plotting of both images and curves on a single graph,
3. must allow display of multiple graphs at once,
4. must allow return of arbitrary information to calling program,
5. should have an intuitive GUI interface, and
6. should be built with widely supported free software.

© Copyright 1997 Astronomical Society of the Pacific. All rights reserved.

3. Implementation

We used the existing Tk extension VISU for our image display engine, and cannibalized existing plotting code written for the *fv* project for the curve plotting and axis drawing. All "drawing" is done as "items" on a Tk "canvas." The result is a simple and powerful tool which fits smoothly into the highly customizable Tcl/Tk environment.

The power of Tk then allows for a very general interface to receive input from the user. Tk allows the developer to query raw cursor positions on the canvas. POW provides the developer with several utilities for translating cursor positions into physical coordinates and in which graph a given cursor position falls. This allows the developer to write whatever sort of interface makes the most sense for his application, using simple Tk code.

4. Structure

All elements in POW may be thought of as "Objects" (see Table 1)

Table 1. The object types in POW.

Object	Created from
Data	a (void) pointer to some data
Vector	Data plus units
Curve	2 to 6 Vectors
Image	Data plus origin, size of pixels, and units
Graph	List of Images and Curves (this is the Displayed object)

5. GUI

The end user can perform many activities on displayed Graphs via the POW GUI:

- graph objects can be moved around, zoomed, magnified, and panned,
- individual items (lines, text, images) can be deleted or changed from the GUI,
- applications programmers or end users (via a dialog box) can add features by sending arbitrary scripts to the Tcl interpreter, and
- the POW display can be dumped to a file in PostScript or PPM formats.

6. POW "Methods"

These methods are implemented as C functions, TCL procs, or FORTRAN subroutines (via the black magic of cfortran.h). (The C/FORTRAN versions carry a status variable to fit FTOOLS conventions.)

6.1. Object Constructors

- PowInit(): Creates a POW window.
- PowCreateData(data name, data array, data type, length, copy): Passes a pointer to some data to POW and gives it a name. Pow can make its own copy of the data or use yours, whichever best suits your memory management needs.
- PowCreateVector(vector name, data name, offset, length, units): Gives 1-D physical meaning to a chunk of Data. There is also a function to create a vector and its associated data given a start value and an increment.
- PowCreateCurve(curve name, x vector, x error, y vector, y error, z vector, z error): combine vectors into a Curve.
- PowCreateImage(image name, data name, xoffset, yoffset, width, height, xorigin, xinc, yorigin, yinc, xunits, yunits, zunits): Give 3-D physical meaning to a chunk of Data.
- PowCreateGraph(graph name, curves, images, xunits, yunits, xlabel, ylabel, xdim display, ydim display, xmin, ymin, xmax, ymax): Display a list of curves and images as a Graph on the POW canvas. Most of these parameters are optional and will default to "sensible" values if left NULL.

6.2. General Utilities

This is only a partial list.

- powPlotCurves(graph, curves): Adds the list of Curves to an existing Graph.
- powPlotImages(graph, images): Adds the list of Images to an existing Graph
- powMagGraph(graph, newmagstep): Resizes a Graph to a given magstep (magstep = 1 is the "natural" size of the graph. Magsteps must be integers or 1/integers).
- powStretchGraph graph factor: Shrinks or expands a Graph to the allowed magstep nearest to (current magstep $\times$ factor).
- powFindCurvesMinMax(curves, axis): Takes a list of curves and an axis and returns the minimum and maximum values of the curve along that axis.

6.3. Cursor Positions

The developer can bind mouse clicks (or other X events) to send him cursor positions. The following routines can then make sense of the positions.

- powGraphToCanvas(graph, axis, coordinate): Takes a physical coordinate and returns the corresponding position on the POW canvas.
- powCanvasToGraph(graph, axis, coordinate): Takes a canvas coordinate and returns the corresponding physical coordinate. The second argument specifies x or y axis.
- powWhereAmI(x, y): takes an (x, y) pair of canvas coordinates and returns which graph (if any) in which they fall.

6.4. Linked Axes

POW allows you to "link" together any number of axes on different graphs. The resulting set of linked axes is called a "chain." Each axis can be a member of only one chain. Linking an axis from one chain to an axis in another chain has the effect of merging the two chains. Zooming on a region of interest on one graph will affect the linked axis on any other graph. There are several utility routines. Also, the GUI allows the user to view links as lines connecting linked axes.

- powLinkAxes(graph1, axis1, graph2, axis2): Links two axes.
- powBreakLink(graph, axis): Removes the specified axis from its chain.
- powAlignChain(graph, axis, orientation): Moves all graphs belonging to the same chain as the specified graph so that they are aligned on the canvas (i.e., it "stacks" the graphs into a column or lines them up in a row on the user's screen).

7. Availability

An early version of POW is integrated into *fv* in the FTOOLS3.6 distribution and the standalone *fv* distribution. A much improved version of the POW library should be available by the time these proceedings are published. All of this software is available from the HEASARC Web site[1] under the link labeled "Software".

"The road of excess leads to the Palace Of Wisdom"
The Marriage of Heaven and Hell
William Blake

---

[1] http://heasarc.gsfc.nasa.gov/