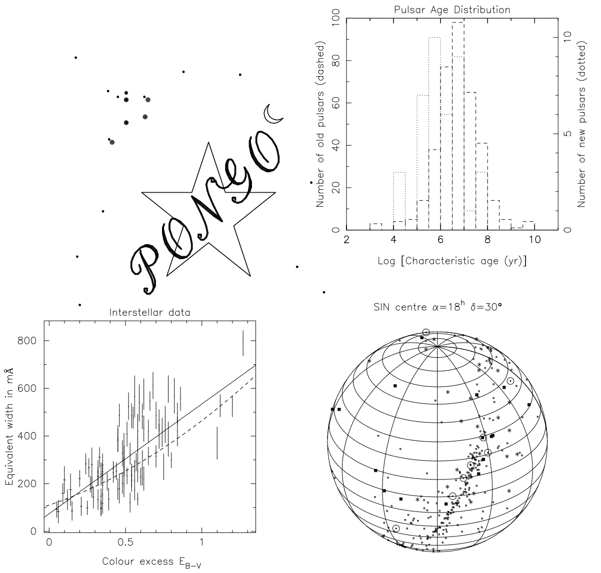
SUN/137.7

Starlink Project
Starlink User Note 137.7

Paul Harrison, Paul Rees,
Peter Draper, Alasdair Allan

27 March 2008

PONGO — A Set of Applications for Interactive Data Plotting
2.0-3
User's Manual



Abstract

PONGO is a suite of commands for interactively plotting data stored in text files.

Contents

List of Figures

1 Introduction

PONGO is a set of commands for interactively plotting data stored in ordinary text files. It is based around the PGPLOT (SUN/15) graphics package.

Some highlights of PONGO are:

- Data are read from text files using a command (READF) that has:
    - the ability to read files which contain character strings as well as numeric values;
    - support for comment lines and column headings as an aid to remembering what the file contains;
    - variable column delimiters, allowing LaTeX and TeX format tables to be read;
    - selective reading of the data;
    - the ability to read in positions of Right Ascension and Declination in the format

      ```
      HH:MM:SS.SSS   +DD:MM:SS.SSS
      ```
- Plots of astronomical positional data can be made in one of several geometries (TAN, SIN, ARC, GLS, AITOFF, MERCATOR and STG).
- Complicated mathematical manipulations can be performed on the data using Fortran-like statements to define the required transformation.
- Specialised extra data columns are provided, *i.e.* LABCOL for labels, SYMCOL for symbol numbers and EXCOL/EYCOL for errors.
- Many interactive cursor functions are provided.
- Error ellipses can be drawn.
- Vector plots can be drawn.
- Simple statistical analysis of the data is available.
- The data can be resampled.
- Functions defined by Fortran-like statements can be drawn.

PONGO can used from both the Starlink/ICL and IRAF/CL [1] command languages and integrates with other Starlink software packages.

Because the parameters PONGO uses for its commands are often similar to the arguments of the equivalent PGPLOT subroutines, it is useful to read the PGPLOT manual in conjunction with this user guide (if you are not already familiar with PGPLOT). In several cases the full descriptions of the parameters are given only in the PGPLOT manual.

[1] at the time of writing IRAF support is no longer available in a standard release

2 Starlink/ICL or IRAF/CL?

PONGO needs a command-language as a host environment and at present it can be used from two that are generally available to astronomers; the Starlink Interactive Command Language (ICL) and the IRAF Command Language (CL). You should use PONGO from which ever language is most natural (*i.e.* the one you or your colleagues are most familiar with). There are no significant advantages to either system, however, people intending to use IRAF/CL should note that most of this document is slanted towards ICL usage (the original "home" of PONGO) and should be prepared to consult the on-line help system, which should be more IRAF specific.

Under IRAF/CL PONGO is available as the pongo package. From ICL PONGO is available by just typing the pongo command.

3 Tutorial Examples

The following two sections show the same simple example. The first uses the Starlink/ICL and the second the IRAF/CL command languages.

3.1 An ICL example

In order to start PONGO you must first start ICL. This may be done by typing:

```
% pongo
```

at C-shell the prompt. Or alternatively by starting ICL then using the pongo command.

In any PONGO session the first action is to open the plotting device. This is done by typing[2]:

```
ICL> begplot xw
```

Where 'xw' should be replaced by the name of the plotting device you are intending to use. Note that once you have begun a PONGO session with the BEGPLOT command, the ICL> prompt changes to PONGO>. This is important because the only commands you can use with success at the ICL> prompt are BEGPLOT and DEVICE, all other commands should be typed at the PONGO> prompt. The data may then be read using the command [3]. Normally an 'ordinary' file name would be used *i.e.* spectrum.dat. An ICL variable is used here as it is not possible at this time to use an environment variable for accessing formatted files. The actual file is $PONGO_EXAMPLES/tutorial.dat:

```
PONGO> readf data=(TUTORIAL_DATA) xcol=1 ycol=3 all
```

[2] If you have already used PONGO then you should also issue the RESETPONGO command immediately after BEGPLOT.

[3] Note that the data file name is given as (TUTORIAL_DATA), this uses an ICL variable which is set up when PONGO is initialised

Plotting limits are set up using the range of the data by the command:

```
PONGO> dlimits
```

Axes for the plot may be drawn using:

```
PONGO> boxframe
```

and finally the points may be plotted as asterisk-like symbols and the axes labelled with:

```
PONGO> points 3
PONGO> label 'X axis' 'Y axis' 'Plot Title'
```

Note that the data values are remembered by PONGO and the plot you have just created may be erased and recreated by typing:

```
PONGO> advance
PONGO> boxframe
PONGO> points
PONGO> label
```

To close a device and end a PONGO plot the command ENDPLOT should be used. This is important if you are going to switch to another set of applications such as KAPPA, otherwise the plotting device characteristics will be inaccessible to the second package.

3.2 A CL example

In order to start PONGO you must first start CL and load the pongo package.

In any PONGO session the first action is to open the plotting device. This is done by typing[4]:

```
po> begplot xw
```

Where 'xw' should be replaced by the name of the plotting device you are intending to use. The data may then be read using the command:

```
po> readf data=pongoexamples$tutorial.dat xcol=1 ycol=3 all=yes
```

Plotting limits are set up using the range of the data by the command:

```
po> dlimits
```

Axes for the plot may be drawn using:

```
po> boxframe
```

and finally the points may be plotted as asterisk-like symbols and the axes labelled with:

```
po> points 3
po> label "X axis" "Y axis" "Plot Title"
```

To close a device and end a PONGO plot the command: ENDPLOT should be used.

[4] If you have already used PONGO then you should also issue the RESETPONGO command immediately after BEGPLOT.

4 Classified List of Commands

This section presents a list of the available PONGO commands, classified into several broad categories: commands which begin and end a PONGO session, commands for manipulating data, commands which control plotting, plotting commands, and commands which perform simple statistics on the data. Not all the commands given are individual applications, many are synonyms (or scripts) for other applications with specific parameters provided for convenience. Detailed descriptions of these commands are given in Appendix B. The parts of the command names outside parentheses define the minimum abbreviation for that application (IRAF users should note that CL uses its own abbreviations, which may not correspond to those below).

4.1 Commands which begin and end PONGO

BEGP(LOT) – Open a plotting device.

DEVICE – Open a plotting device.

ENDP(LOT) – Close down the current plotting device.

4.2 Commands for plotting

ADV(ANCE) – Clear the graphics screen.

ANN(OTATE) – Annotate the plotted data.

ARC – Draw an arc of an ellipse.

BIN – Plot a histogram using previously binned data.

BOX(FRAME) – Draw a frame and axes on the plot.

CONN(ECT) – Draw straight lines between the data points.

CURSE – Use the cursor to get interactive input.

DRAWPOLY – Draw a polygon.

DRAW – Draw a line from the current pen position the specified point.

ELLIPSES – Draw error ellipses.

ERASE – Clear the graphics screen

ERRORBAR – Draw error bars on the plotted data.

ERRX – Draw symmetric error bars in the X direction.

ERRY – Draw symmetric error bars in the Y direction.

GPOINTS – Plot points or lines between the data.

GRID – Draw a coordinate grid at specified intervals.

GT_CIRCLE – Draw a great circle between two points.

HIST(OGRAM) – Bin and plot a histogram of the data.

LAB(EL) – Draw the axis labels and title on the plot.

MARK – Draw a point mark at the specified position.

MTEXT – Draw a text string relative to the viewport.

PLOTF(UN) – Plot a given function.

PLOTH(IST) – Plot a histogram of the data.

POI(NTS) – Draw a point mark at each of the data points.

PRIM – Perform primitive plotting functions.

PTEXT – Draw a text string at the specified position and angle.

PVECT – Draw proper motion vectors.

RADIATE – Draw a line from the given point to the first NP data points.

SIZE(PLOT) – Draw point marks of differing sizes at each of the data points.

TEXT – Draw a text string on the plot at the specified position.

VECT – Draw vectors from each data point.

WTEXT – Draw a text string on the plot.

XERR – Draw symmetric error bars in the X direction.

YERR – Draw symmetric error bars in the Y direction.

4.3 Commands which control plotting

CHANGE – Change plotting attributes.

CLEAR – Clear plotting attributes.

DLIM(ITS) – Set the world coordinate limits using the data range.

EXPA(ND) – Set the character height.

FILLSTY – Change fill-style plotting attributes.

FONT – Set the text font.

INQ(UIRE) – Display PONGO status information.

LIM(ITS) – Set the world coordinate limits.

LT(YPE) – Set the line style.

LWE(IGHT) – Set the line width.

MOVE – Set the current pen position.

PALET(TE) – Change the plotting pen colours.

PAPER – Change the size and aspect ratio of the plotting surface.

PEN – Set the current pen.

RESETP(ONGO) – Reset the state of PONGO.

SETPROJ – Set the projection geometry.

SHOWP(ONGO) – Show the PONGO global parameters.

VIEWPORT – Set the viewport for the current plotting device.

VPORT – Set the viewport using normalised device coordinates.

VP_BH – Set the viewport to the bottom half of the plotting surface.

VP_BL – Set the viewport to the bottom-left quarter of the plotting surface.

VP_BR – Set the viewport to the bottom-right quarter of the plotting surface.

VP_TH – Set the viewport to the top half of the plotting surface.

VP_TL – Set the viewport to the top-left quarter of the plotting surface.

VP_TR – Set the viewport to the top-right quarter of the plotting surface.

VSIZE – Set the viewport using its physical size in inches.

VSTAND – Set the standard viewport.

WNAD – Adjust the viewport so that the X and Y scales are the same.

WORLD – Set the world coordinates for the plot.

4.4 Commands for manipulating data

AVEDAT – Average the data in the XCOL and YCOL areas.

CCMATH – Perform inter-column maths.

CLOG – Take the logarithm of a column.

DATA – Specify the data file name.

DEGTOR – Convert the specified data area from degrees to radians.

EXC(OLUMN) – Specify the column containing the X-axis error data.

EYC(OLUMN) – Specify the column containing the Y-axis error data.

GETP(OINT) – Retrieve information for a plotted data point.

LABC(OLUMN) – Specify the column used for data labels.

PCOL(UMN) – Specify the column used for symbol codes.

PTINFO – Get the coordinates of a specified data point.

READ(F) – Read from a formatted data file.

RTODEG – Convert the specified data area from radians to degrees.

SYMC(OLUMN) – Specify the column used for symbol codes.

WRITE(I) – Write information to an output file.

XC(OLUMN) – Specify the column containing the X-axis data.

XLIN(EAR) – Put 1. . . N into the XCOL data area.

XLOG(ARITHM) – Take the logarithm of the X-axis data.

XOFF(SET) – Add a constant offset to the X-axis data.

XSCALE – Multiply the values in the XCOL and EXCOL data areas by a constant.

YC(OLUMN) – Specify the column containing the Y-axis data.

YLIN(EAR) – Put 1. . . N into the YCOL data area.

YLOG(ARITHM) – Take the logarithm of the Y-axis data.

YOFF(SET) – Add a constant offset to the Y-axis data.

YSCALE – Multiply the values in the YCOL and EYCOL data areas by a constant.

ZC(OLUMN) – Specify the column containing the Z-axis data.

ZSCALE – Multiply the values in the ZCOL data area by a constant.

4.5 Commands for performing simple statistics

FITC(URVE) – Fit a curve to the data.

FITL(INE) – Fit a straight line to the data.

5 Data File Formats

The simplest form of input file for PONGO is a text file with the data in columns separated by spaces, as in the file `$PONGO_EXAMPLES/tutorial.dat` (or `pongoexamples$tutorial.dat`). However, PONGO allows a considerable level of fine control over the data used for plotting from a particular file by providing for the use of column delimiters, column labels and comments in data files.

5.1 Column delimiters

The default column delimiter is a space character, although the `READF` command does have the ability to use other delimiters by setting the `DELIM` parameter. It is possible for more than one delimiter character to be used, *e.g.* using `&\` would be a good way to read a table that was in TeX format (`&` for a LaTeX tabular format table). A null string for the `DELIM` parameter has the same effect as a single space.

5.2 Column labels

It is possible to give each column in a file a symbolic name that can be used to reference the column when reading the file and can be automatically transferred to the appropriate axis label on the plot. To do this, the first line of the file should be of the form:

```
!$label 1$label 2$label 3$
```

where there are as many labels, each delimited by a `$`, as there are columns. Care should be taken to ensure that there is no leading white space in the column labels, although it is permissible for the column labels to contain white space elsewhere. Any padding that is required to make the column labels line up with the data columns should be achieved with multiple dollar signs, *e.g.* :

```
!$RA$$$$Declination$
```

When specifying the column on the command line, *e.g.* `YCOL Dec`, it is permissible to abbreviate the string to a minimum match. However, *the match is case-sensitive.*

5.3 Comments

Comments may be placed in the data file by prefixing the line with one of the standard comment characters. There are two comment characters allowed in the data file, specified by the parameters `HARDCOMM` and `SOFTCOM` for the command `READF`. These comment characters must occur in the first column of a line to be recognised as comment characters. The main purpose of comment characters is to document data files and to comment out unwanted lines of data. The existence of two comment characters provides the ability to selectively read data subsets from files. Blank lines are ignored in data files.

Although not recommended practice, the documentation of data files can be done without the use of comment characters: because PONGO will reject any line in which the required numerical column cannot be interpreted as a valid number, it does not matter whether a comment character is put at the start of the line or not.

5.4 Astronomical coordinates

It is possible to read data stored in an astronomical coordinate format (*i.e.* `HH:MM:SS.SSS` and `DD:MM:SS.SSS`) into PONGO. The data is stored internally in *radians*: in this conversion any data read into XCOL is assumed to be a Right Ascension (*i.e.* of the first form above) and any data read into YCOL is assumed to be a Declination (*i.e.* of the second form above).

The `BOXFRAME` command can label axes using a `HH MM SS.S` and `DD MM SS.S` format providing the data and frame limits are specified in radians.

6 Internal Data Areas

There are seven internal data areas into which data may be read. These are as follows:

XCOL – This area is principally intended to hold X-axis data. However, when using the command `PLOTHIST H`, this area is intended to hold the data before they are binned to draw the histogram.

When using the command `ERRORBAR X` with the symmetric option turned off, this area contains the position of one of the ends of the error bar.

YCOL – This area is intended to hold Y-axis data.

When using the command `ERRORBAR Y` with the symmetric option turned off, this area contains the position of one of the ends of the error bar.

EXCOL – This area is principally intended to hold errors for the X-axis data values. These errors are generally assumed to be symmetric about the data point by the command `ERRORBAR`, but with the symmetric option turned off EXCOL holds the position of the opposite end of the error bar to XCOL.

The EXCOL area is assumed to contain the standard error of the XCOL data by the command `ELLIPSES`.

The EXCOL area is also used to hold the X-axis component of the vector offset from the data point by the command `VECT`, and the proper motion in Right Ascension by the command `PVECT`.

When using the command `ARC`, the EXCOL data are assumed to contain the magnitude of the semi-major axis of the ellipse.

The command `AVEDAT` will write the standard deviation of the average it calculated for each bin into the EXCOL data area.

EYCOL – Similar to EXCOL, but applying to the Y-axis data values.

The EYCOL area is used to hold the Y-axis component of the vector offset from the data point by the command `VECT`, and the proper motion in Declination by the command `PVECT`.

When using the command `ARC`, the EYCOL data are assumed to contain the magnitude of the semi-minor axis of the ellipse.

ZCOL – PONGO does not have the ability to draw three dimensional plots. As a result, the ZCOL data area does not contain Z-axis data in the conventional sense. However, there are a number of commands that use the values stored in the ZCOL area.

The command `SIZEPLOT` uses the values stored in ZCOL to scale the size of the marker symbols plotted for each data point.

The commands `VECT` and `PVECT` can optionally use the values in ZCOL to scale the length of the vectors they draw.

The command `ELLIPSES` assumes that ZCOL contains the values of the normalised covariances between the X- and Y-axis errors.

LABCOL This area is used for storing a character string associated with each of the data points. The strings may be plotted interactively using the `ANNOTATE` or `CURSE` commands.

SYMCOL This area is used to store a PGPLOT symbol number (an INTEGER) associated with each of the data points.

7 Projections

PONGO is capable of plotting astronomical coordinate data in one of several 'projections' or, more strictly, in one of several geometries (not all of the 'projections' are true projections onto a plane, but are geometries that have other properties – *e.g.* the equal area property where areas are preserved in the transformation from the celestial sphere to a two dimensional plot). *Table 1* lists the geometries available in PONGO.

All the main point plotting commands will work in any of the available geometries. They assume that the data in the XCOL and YCOL areas are stored internally in radians and refer to the longitude and latitude respectively of the point to be plotted. Where the position values are to be entered on the command line, the latitude and longitude are normally given in degrees.

In addition to the point plotting commands, there are also three commands whose only use is when plotting in a particular projection type:

GRID – Plot a coordinate grid in the current projection. The default parameters of this command are to draw a grid over the whole sky with a longitude line every 30 degrees and a latitude line every 10 degrees.

GT_CIRCLE – Draw a great circle between two specified points on the celestial sphere.

PVECT – Draw proper motion vectors from each plotted point as small great circle arcs.

8 Creating Complex Plots

Repeating complex sequences of PONGO commands by hand can become boring and is likely to lead to many mistakes (particularly on those occasions when you want to create a hardcopy and cannot even see the current state). The way to get around these problems is to use the abilities of the ICL or CL command languages to create *procedures*.

The following two sections show extremely simple examples of how to start doing this in both languages. To get further you'll need to consult the appropriate documentation, SG/5 in the case of ICL and the "*CL Programmer's Manual*" and "*An Introductory User's Guide to IRAF Scripts*" for CL.

8.1 A simple ICL procedure

A simple procedure that reproduces the tutorial example looks like:

Projection	Description
NONE	No projection is used.
AITOFF	The Aitoff projection (has an equal area property).
ARC	Move an equal distance on the tangent plane as a great circle on the celestial sphere.
GLS	Right Ascension intervals are multiplied by $\cos\delta$ (has an equal area property).
MERCATOR	The Mercator projection.
SIN	Drop a perpendicular onto the tangent plane.
STG	The Stereographic projection.
TAN	Projection onto the tangent plane through the centre of the celestial sphere.

Table 1: Projection geometries available in PONGO

```
proc icltest file
   begplot xw
   readf data=(file) xcol=1 ycol=3 all reset
   dlimits
   boxframe
   points 3
   label ’X axis’ ’Y axis’ ’PLOT TITLE’
   endplot
endproc
```

If you enter this text into a file say called `icltest.icl` and use the following commands from ICL:

```
ICL > load icltest
ICL > icltest (TUTORIAL_DATA)
```

The data from the tutorial example will be plotted as before. One immediate advantage of this method is that we can now plot data from columns 1 and 3 of any file.

If the contents of the procedure are wrong or need a slight modification then you can edit the procedure from ICL. Try the command:

```
ICL > edit icltest
```

This puts you into an editor and you can modify the procedure. (If you haven’t set an environment variable EDITOR or used the ICL command `set editor`, you’ll find yourself in `vi`. To exit from this type the command `:q` and now use the `set editor` command.) If you made a modification to the procedure this can be viewed using the command:

```
ICL > list icltest
```

All the currently available procedures are shown using the command:

```
ICL > procs
```

If you do modify a procedure from ICL use the:

```
ICL > save icltest
```

command to write out your changes (you should use the name of your procedure instead of `icltest`).

8.2 A simple CL procedure

A simple CL procedure that reproduces the tutorial example looks like:

```
procedure cltest (datafile)
string datafile
begin
   begplot xw
   readf (data=datafile, xcol=1, ycol=3, all=yes, mode=h )
   dlimits
   boxframe
   points 3
   label ("X axis", "Y axis", "PLOT TITLE")
   endplot
end
```

If you enter this text into a file say called `cltest.cl` and use the following commands from CL.

```
po> task cltest = cltest.cl
po> cltest pongoexamples$tutorial.dat
```

The data from the tutorial example will be plotted as before. One immediate advantange of this method is that we can now plot data from columns 1 and 3 of any file.

Another way in which you can create CL scripts is to use the `mkscript` command.

8.3 Example procedures

These examples are intended to show some of the features of PONGO and also provide examples of complex procedures for both the ICL and CL command-languages. All these procedures can be found in the directories `$PONGO_EXAMPLES` and `pongoexamples$` respectively. ICL examples have file extensions `.icl` and CL examples `.cl`.

The ICL procedures will prompt for a graphics device and should be invoked from the `ICL>` prompt using the ICL command `load`, *e.g.* :

```
ICL> load $PONGO_EXAMPLES/ppdotdiag
```

and their execution can be followed if the ICL command `set trace` has previously been executed.

The CL procedures are defined as tasks by the `pongo` command and can be run by just typing their name *i.e.* :

```
      po> ppdotdiag
```

Note that some of the examples involve the colour representation of lines, which may be difficult to see on a greyscale device (*e.g.* on a monochrome Postscript device). It is recommended that a colour image display device is used to run the example procedures.

8.3.1 Spectrum

This procedure produces *Figure 1*, a plot of a low resolution IUE spectrum extracted by IUEDR (SUN/37) and written using the IUEDR command `OUTSPEC SPECTYPE=2`. The output file was subsequently edited to make the file label lines PONGO comments and to add a line of PONGO column labels (see §5) at the beginning of the file. IUEDR indicates bad or missing data in an output spectrum by attributing zero fluxes to the affected wavelength samples. These can be detected and discarded using the `SELCOND` parameter of the `READF` command: *e.g.*

```
PONGO> readf (SWP_DATA) selcond="Flux > 0.0" noall
```

In this example PONGO draws a polyline across all missing data flagged by IUEDR.

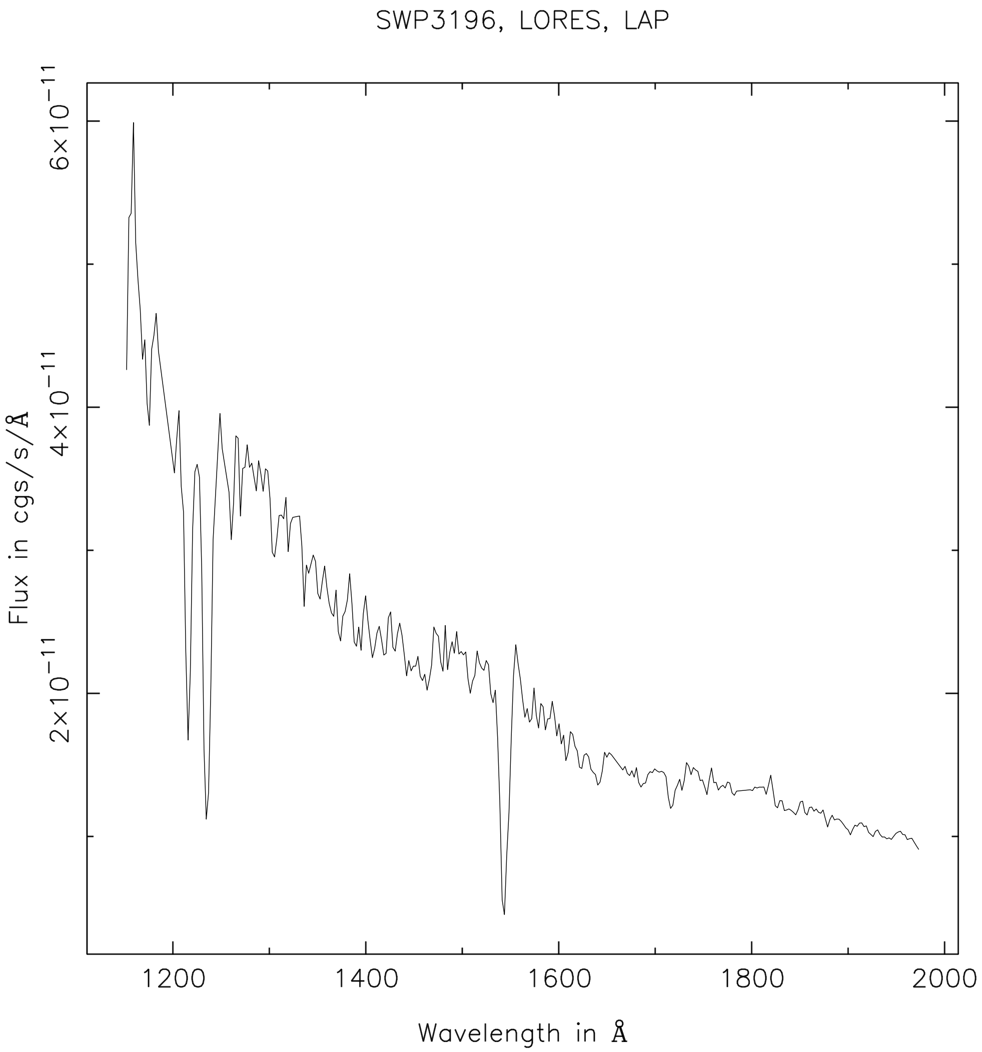


Figure 1: SPECTRUM example.

8.3.2 Errors

The procedure ERRORS was used to plot *Figure 2*. This example demonstrates plotting simple error bars using PONGO (the ERRORBAR command) and also performing simple statistics on the data (the FITLINE and FITCURVE commands). Note that a summary of the statistics is reported for each fit to the data.

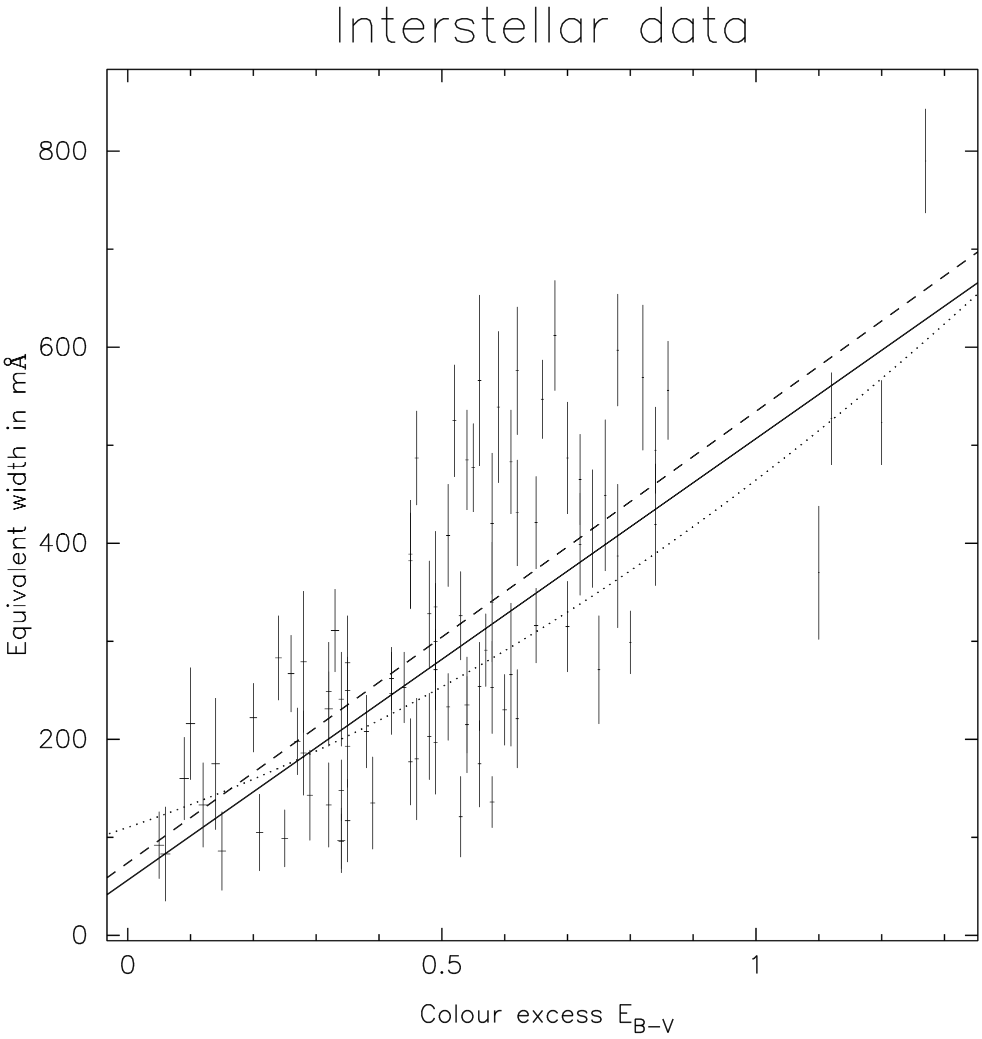


Figure 2: ERRORS example.

8.3.3 Histogram

The procedure HISTOGRAM (histogramtest from CL) was used to plot *Figure 3*, which illustrates the plotting of histograms with PONGO (the PLOTHIST command). This procedure also illustrates how the drawing of the box around the plot can be controlled using the BOXFRAME command.

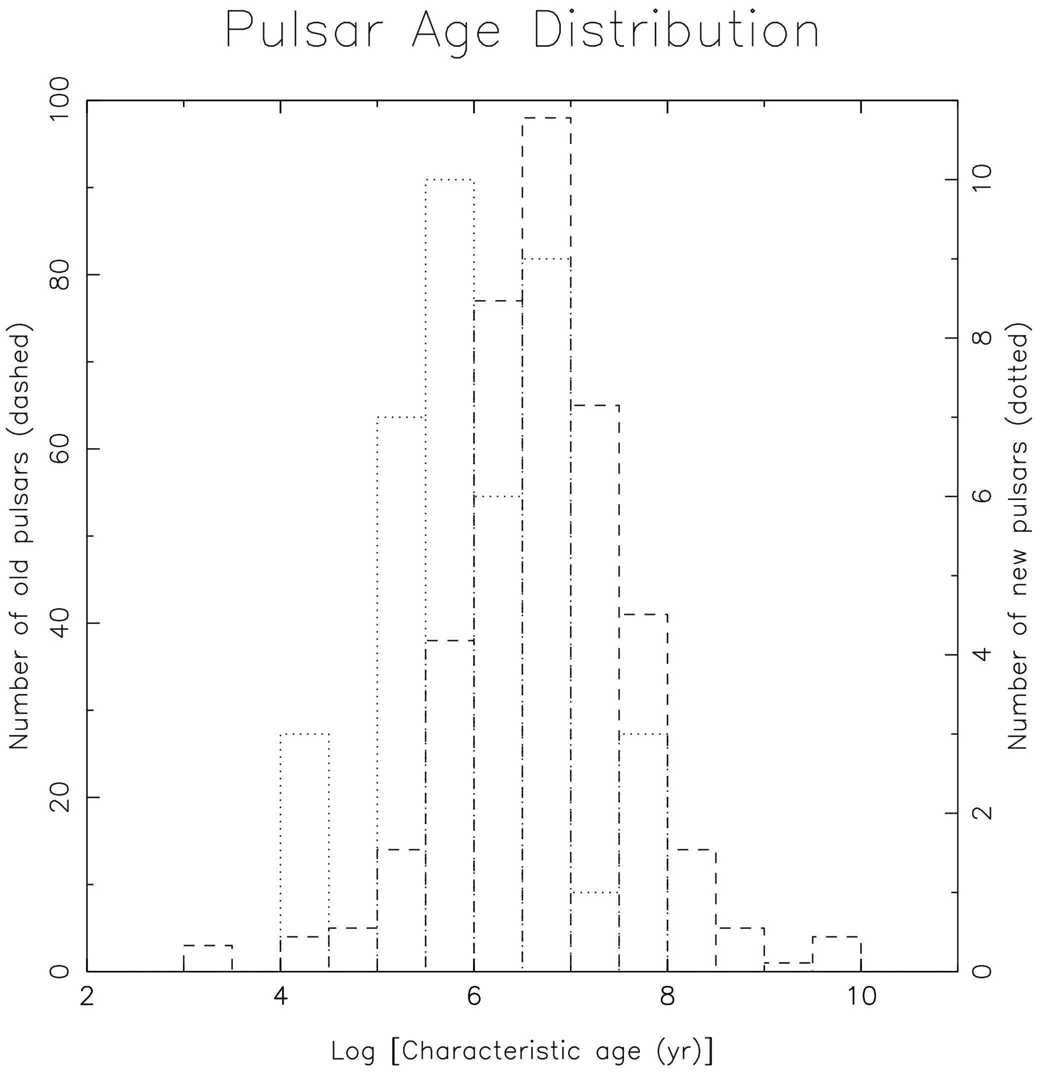


Figure 3: HISTOGRAM example.

8.3.4 Ppdotdiag

This procedure produces *Figure 4*, a period versus period derivative diagram for pulsars. Note the use of a column within the data file to set the symbol number of each plotted point, and the use of the PONGO_NDATA global parameter for making a title containing the number of points that have been read in (in CL this uses the output parameter readf.ndata)

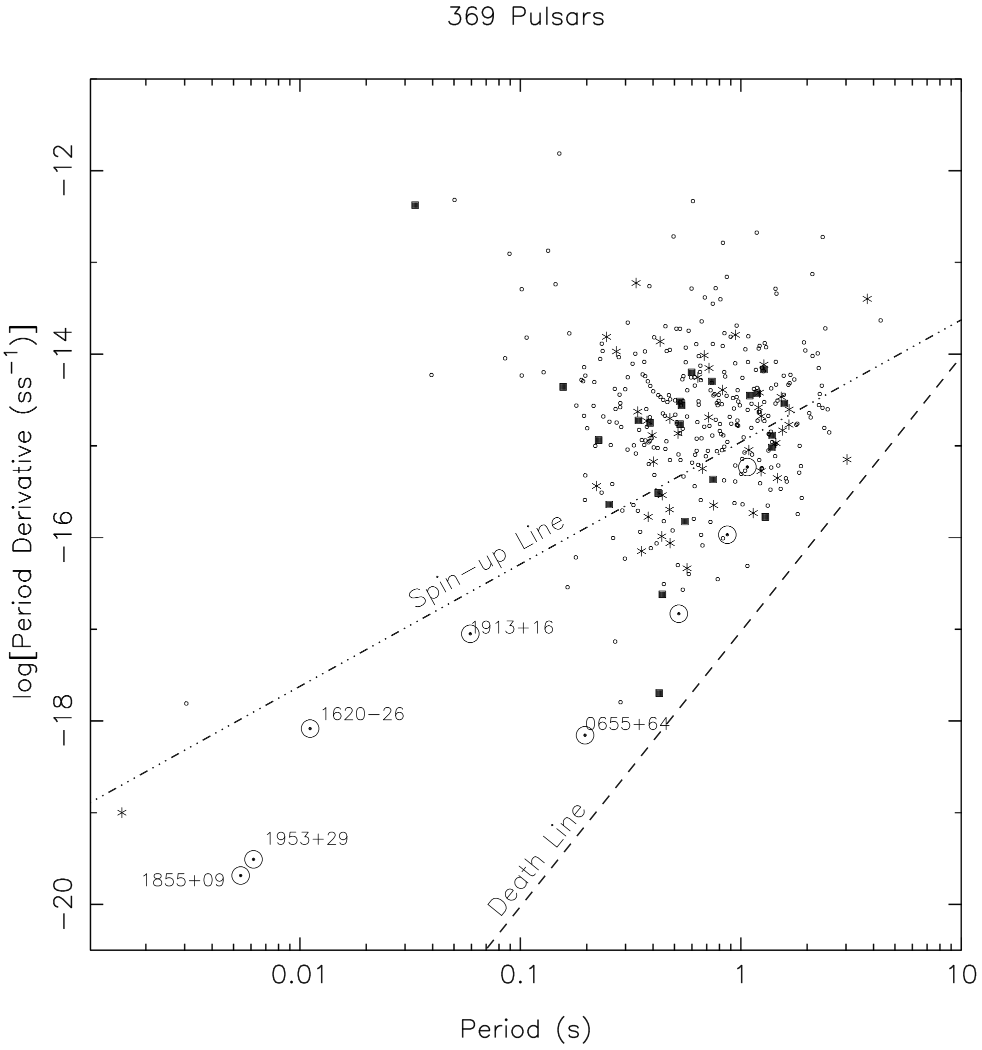


Figure 4: PPDOTDIAG example.

8.3.5 Ellipses

The procedure ELLIPSES (`ellipsetest` from CL) was used to plot *Figure 5*, which illustrates the use of the ELLIPSES command for plotting error ellipses.

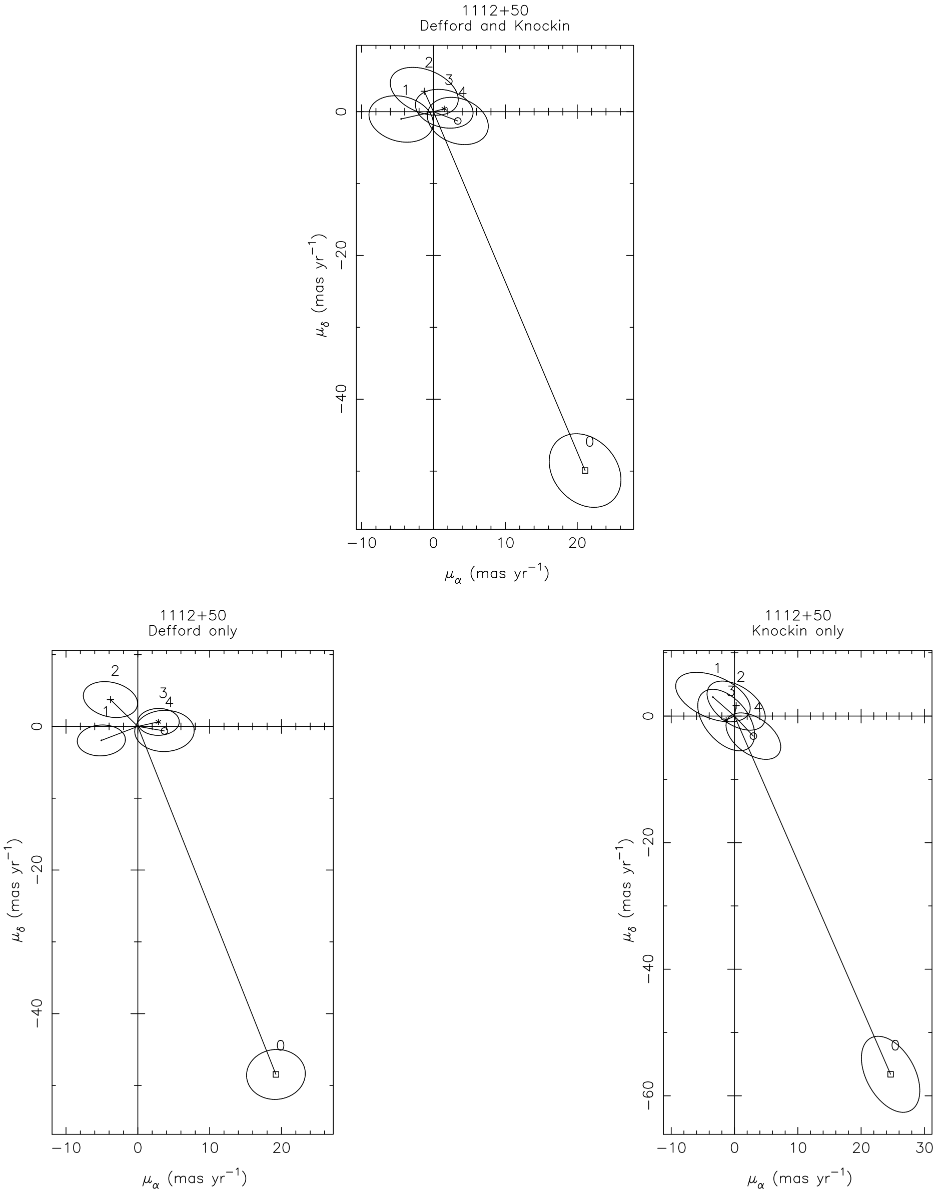


Figure 5: ELLIPSES example.

8.3.6 Projections

This procedure illustrates some of the different ‘geometries’ available in PONGO. It plots four different views of the distribution of a selection of the known pulsars in Right Ascension and Declination to produce *Figure 6*.

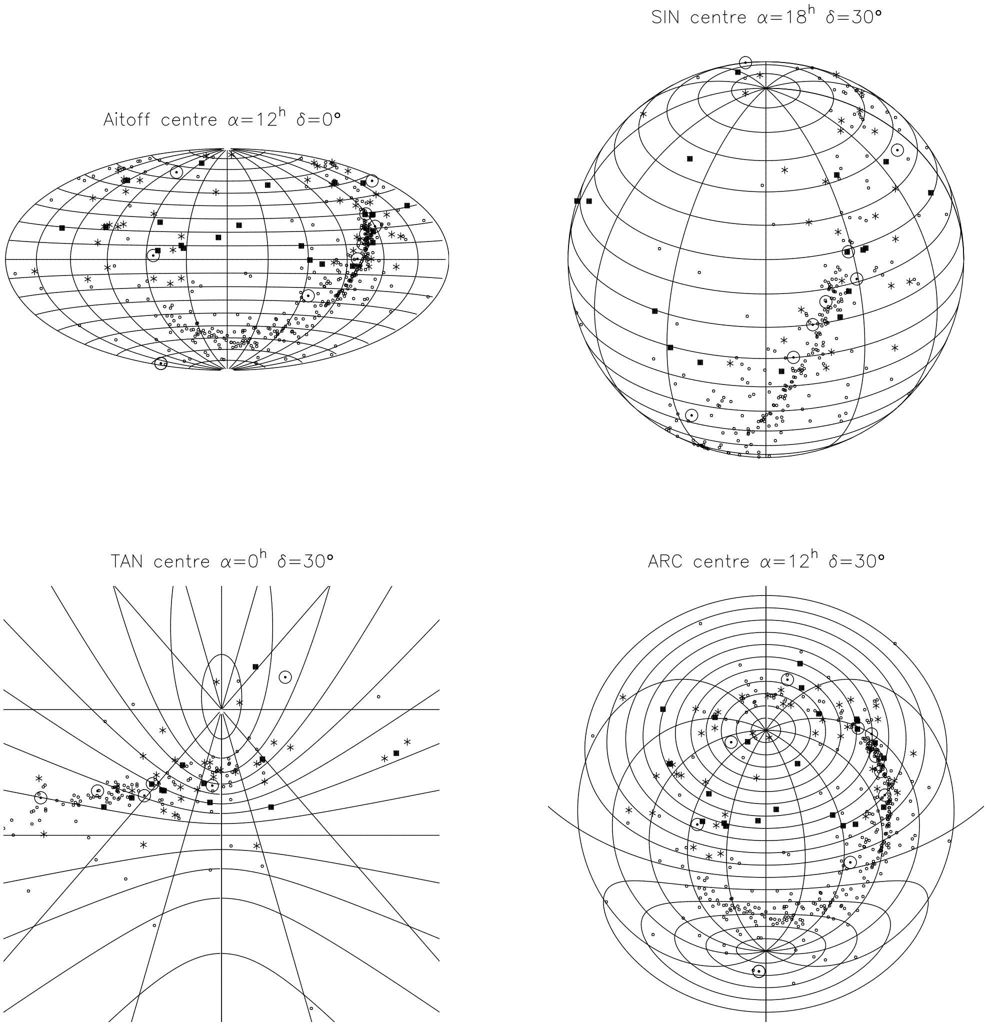


Figure 6: PROJECTIONS example.

8.3.7 Radec

This procedure shows how the `BOXFRAME` command can be used to draw labels in `HH MM SS.S` & `DD MM SS.S` format. The output from it is shown in *Figure 7*. The objects are some of the pulsars from *Figures 6* and *4*.

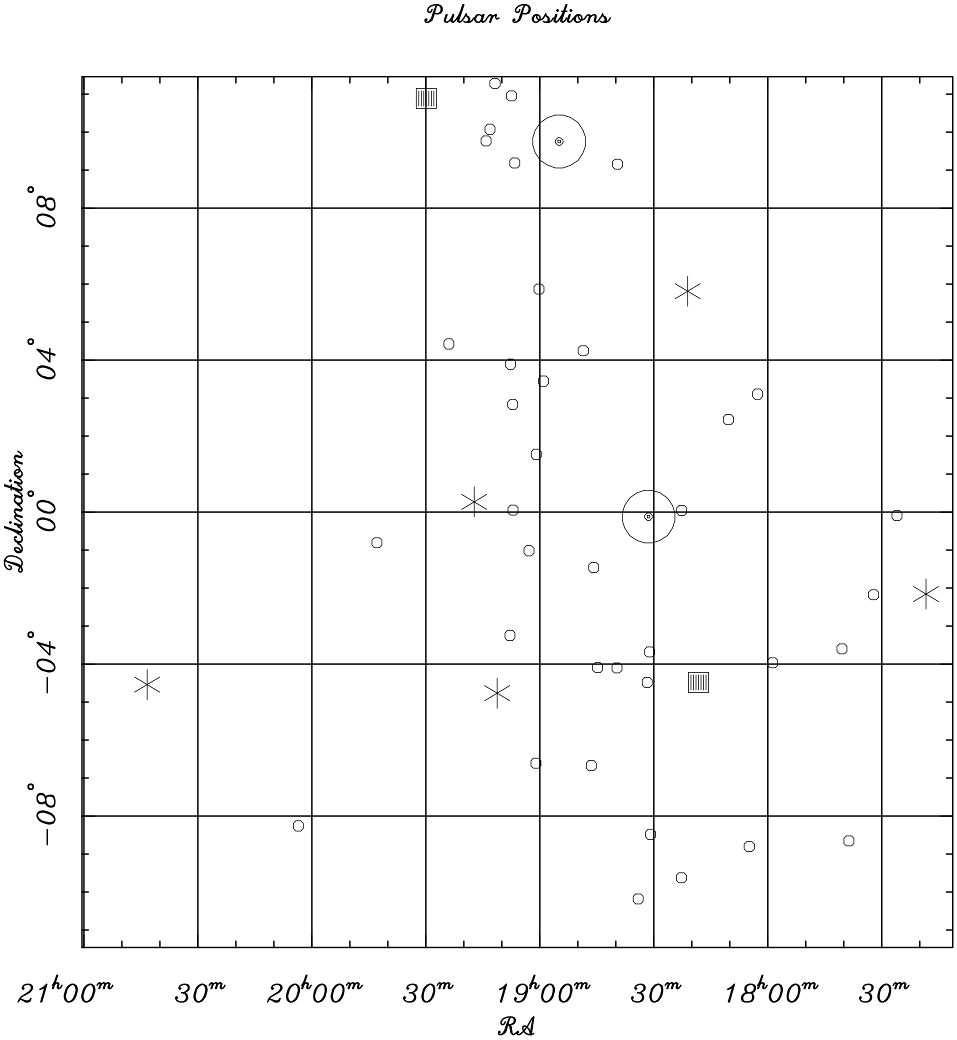


Figure 7: RADEC example.

8.3.8 Vector

This procedure shows the VECT and PVECT commands being used to plot the proper motion vectors of a set of bright stars. The stars are selected to have proper motions in RA greater than 0.5 arcseconds and the distances shown correspond to 100000 years of travel.

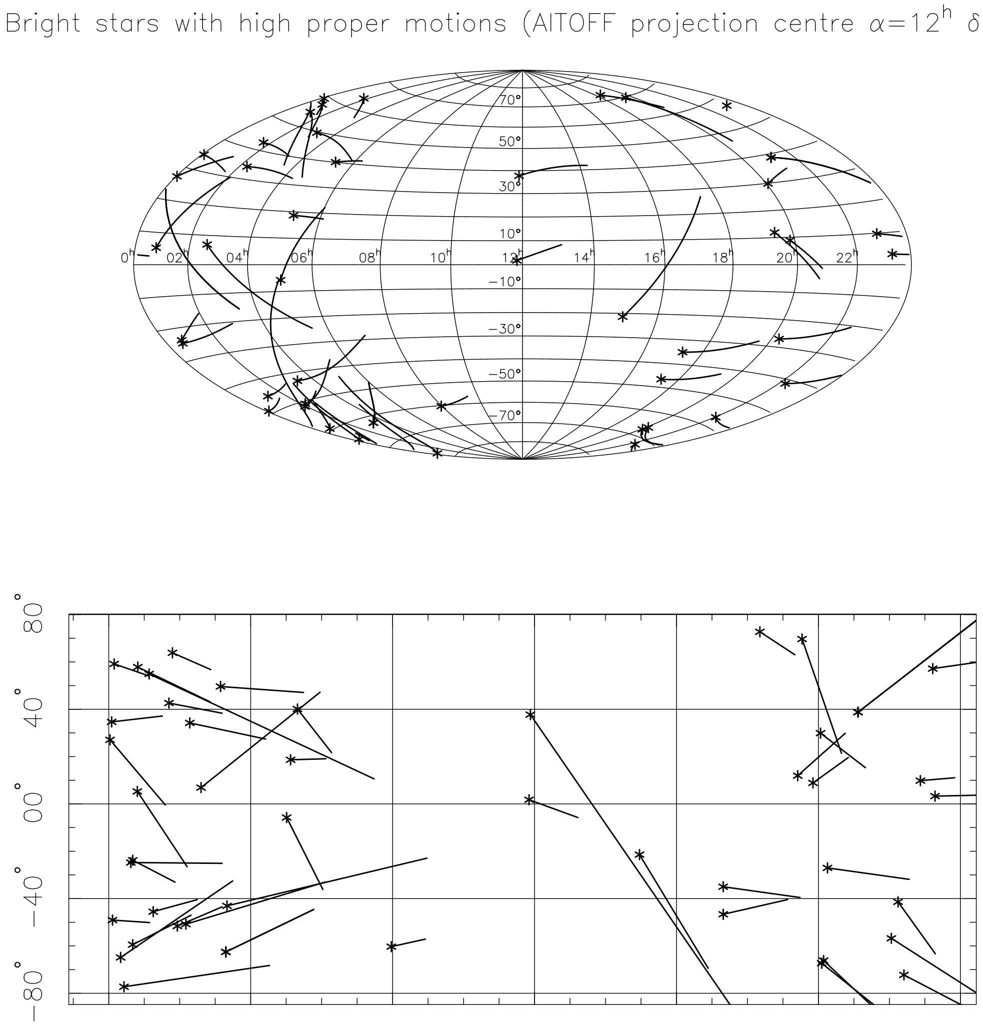


Figure 8: VECTOR example.

8.3.9 Interactive

After executing the examples it is a good idea to gain some experience with the CURSE application. This can be done using the procedure INTERACTIVE. The procedure will plot a graph and then invoke the CURSE application, resulting in the following instructions being printed on the screen:

```
CURSE cursor options:
   Q - Quit.
   D - Draw to the cursor position.
   E - Expand the plotting limits.
   G - Calculate the gradient.
   L - Label the plot.
   M - Mark a point.
   O - Left justified annotation.
   P - Right justified annotation.
   S - Shrink the plotting limits.
   V - Move to the cursor position.
   X - Get the cursor position.
   Z - Zoom the plotting limits.
```

where the letter signifies the key that is to be pressed on the keyboard to achieve the desired effect.

If you have examined the example file for INTERACTIVE you will have seen that a label has been read in for each of the data points. It is now possible for you to use these labels by moving the cursor close to a plotted point and pressing O or P. When you do this, a label for the point nearest the cursor will be written at the cursor position. Do this for several points and then press Q to exit from the CURSE command. PONGO will have remembered the particular labelling that that you have performed and will then use the command WRITEI LABLST=YES to write the PONGO commands required to recreate these labels to a specified file. (You can use the procedure INTERACTIVE to explore the other functions of the CURSE application.)

The ZCOL area has been filled with the distances to each of the pulsars. These values can be used to plot points whose sizes are inversely proportional to distance (*i.e.* the closer pulsars are larger) by first taking the reciprocal of the ZCOL area with the CCMATH command:

```
PONGO> CCMATH Z=1/(Z+0.1)
```

and then the points plotted with the command:

```
PONGO> SIZEPLOT
```

Note here that a small positive value has been added to the distance data (*i.e.* Z+0.1) to avoid very large symbols for nearby pulsars.

8.3.10 AGI

This example illustrates the interaction between the KAPPA package (SUN/95) and PONGO via the AGI graphics database. Here, the image display has been done using the KAPPA `DISPLAY` command and PONGO has been used for all the line drawing (note this example is not available under CL).

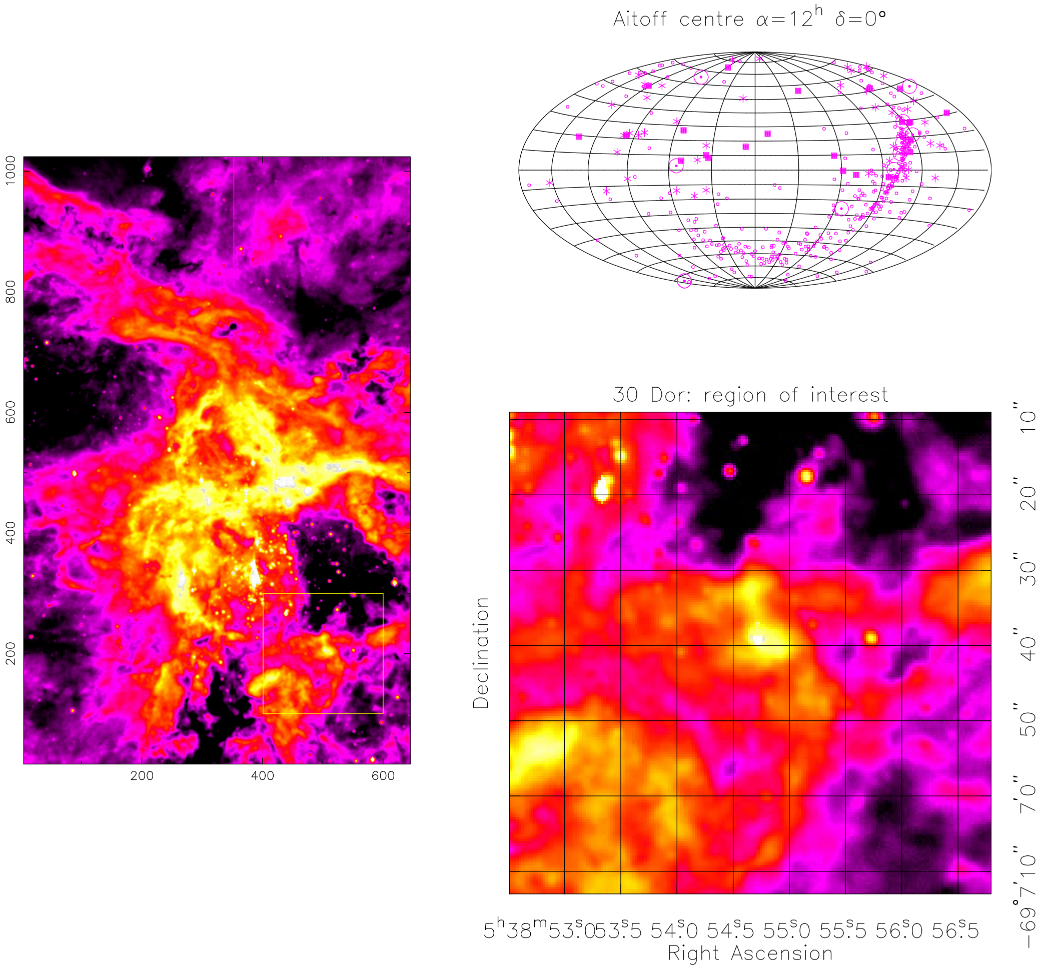


Figure 9: AGI example.

9 Using PONGO with other Starlink Applications

Using graphics between different applications packages is often difficult because once one package has finished plotting and completed execution all information concerning the contents of the plot is lost. This problem may be overcome by storing relevant graphical information for the plot (*e.g.* the graphics device, the size and position of the plot on the display surface, the coordinate limits of the data) in a file and providing facilities to store and retrieve the information in this database. This allows different applications to be used to display images, draw contours and annotate, for example, on the same plot without each application losing access to the plot dimensions. One such facility is provided by Starlink and is called AGI – Applications Graphics Interface (SUN/48). As its name implies, this facility provides a means by which any application can store plotting information for later use and retrieve plotting information stored by previous applications. To use AGI to good effect requires familiarity with some nomenclature.

AGI divides the plotting surface up into **pictures**, where a picture represents a rectangular area on the display surface. AGI stores coordinate information for each picture it creates on a graphics device in its database. AGI pictures are independent of the graphics package used to plot the data (where supported; *i.e.* SGS, IDI or PGPLOT), but they do depend explicitly upon the graphics device used to plot the data. AGI stores each picture in its database, grouped by the graphics device (or **workstation** in Starlink graphics terminology) upon which the picture was created, and each is stored sequentially in the order in which they were created. AGI pictures are given a name which indicates their use: a **BASE** picture refers to the whole plotting surface and the name is reserved only for this use; a **FRAME** picture refers to a picture which contains other pictures; and a **DATA** picture refers to a picture which contains a plot of the data. AGI also stores a label for each picture: a short character string which can be used by an application to uniquely identify a picture. Finally, AGI can store a one-line comment for each picture in the database to describe its use.

PONGO uses the definitions of **world coordinates**, **window**, **device coordinates** and **viewport** used by the PGPLOT graphics package (SUN/15) upon which it is based. Here, a PONGO plot represents a window of given size in the data or world coordinates the you're working in. The data are plotted within these world coordinate limits in a rectangular area of the workstation surface. This area is called the viewport and its size and shape are defined in terms of the device coordinates. The data are plotted exactly within this viewport and so plot annotation and labelling will normally extend beyond the viewport limits. AGI pictures refer directly to PONGO viewports. Clearly, not all viewports used by PONGO will be associated with data windows and refer to plotted data; many will just define regions of the workstation surface in which further viewports will be defined. These viewports are directly analogous to the AGI FRAME pictures and are stored as such by AGI. Those viewports which are used for plotting data within are directly analogous to AGI DATA pictures and again are stored as such by AGI.

By default, when `BEGPLOT` (or `DEVICE`) is used to begin a new PONGO plotting session; *e.g.* :

```
ICL> begplot xw
```

PONGO will open the device and use the AGI BASE picture for that device, unless there is a more recent picture. In the case of a more recent picture, a FRAME picture will be created with the dimensions of the picture current on entry. In either case, the workstation will also be cleared. When a PONGO plot ends, using ENDPLOT, the AGI picture current on entry is again made current.

There are two further modes in which a PONGO plot may be begun: assert that PONGO use the BASE picture for plotting (BEGPLOT ACTION=B); and assert that PONGO use the last DATA picture in the current FRAME, without clearing the graphics device (BEGPLOT OVERLAY NOCLEAR). Specifying BEGPLOT ACTION=B forces PONGO to make the AGI BASE picture current on entry; *i.e.* if you specifically want to use the whole display surface for plotting. Using BEGPLOT OVERLAY=YES CLEAR=NO allows plotting with other applications packages in a coordinated way. Here, PONGO will search for the last DATA picture in the AGI database and use the associated coordinates to define the PGPLOT viewport. The workstation will not be cleared on entry. This means that anything plotted with PONGO will be on the same axes and to the same scale as the existing plot.

As an example, PONGO can be used to annotate images displayed using KAPPA. First, KAPPA will be used to set the image display device and to clear the AGI database for the device (*i.e.* a tidy start):

```
ICL> kappa
ICL> idset xw
ICL> idclear
```

It is necessary to first define the viewport the image is to be plotted within using PONGO in order to assure that there is enough room around the image for the annotation (remember, the annotation of PONGO graphs normally lies outside the viewport):

```
ICL> pongo
ICL> begplot xw
PONGO> vstand
PONGO> endplot
```

Then use KAPPA to display the image[5]:

```
ICL> picdata
ICL> display (DOR_DATA) mode=perc percentiles=[3,99.9]
ICL> lutheat
```

Here, KAPPA PICDATA is used to select the last defined AGI DATA picture, which is a DATA picture created by PONGO with enough space around it to annotate the plot. You could also use KAPPA PICLIST which shows a table of all the pictures in the AGI database. These include the BASE picture, referring to the entire plotting surface of the graphics device; the PONGO FRAME picture, defined on entry and again referring to the entire plotting surface; and finally the PONGO DATA picture, referring the area of the plotting surface within which the image is displayed.

Now PONGO may be used to annotate the plot:

```
ICL> begplot xw overlay noclear
PONGO> boxframe bcinst bcinst
PONGO> label "X-axis pixel" "Y-axis pixel" "KAPPA image"
PONGO> endplot
```

[5] The image actually displayed is defined, in this case, by the ICL variable DOR_DATA. This is set to the value $PONGO_EXAMPLES/dor when PONGO is started.

The image has been displayed by KAPPA and then the axes drawn and annotated using PONGO. In this case, the annotation represents the pixel number in each of the axes of the image. A more detailed example of the interaction between KAPPA and PONGO, via AGI, is provided in §8.3.

10 Panic Section

10.1 Getting help in ICL

When in ICL, on-line help on PONGO may be examined using the command:

```
ICL> help pongo
```

This will provide a brief description of the package and how to begin and end a PONGO plotting session. Once the PONGO commands have been made available within ICL, *i.e.* by typing

```
%  pongo
```

at the C-shell prompt or:

```
ICL> pongo
```

at the `ICL>` prompt, on-line help on any PONGO command may be examined using the command:

```
ICL> help <pongo command>
```

The command:

```
ICL> help pongo
```

will produce a brief introduction to PONGO, and the command:

```
ICL> help introduction
```

will produce a more detailed introduction to PONGO and how to get started. These help commands are also available at the `PONGO>` prompt. A classified list of PONGO applications (`HELP CLASSIFIED`) is provided within the help system and help is also provided on running the PONGO examples (`HELP EXAMPLES`).

If difficulties are encountered with PONGO and the on-line help system does not reveal the cause, it is recommended that the relevant sections of this document are read, or re-read. It may be that this document does not provide enough detail concerning the behaviour of the PGPLOT graphics package: for detailed information regarding PGPLOT, the PGPLOT manual should be consulted (available as a Miscellaneous User Document, MUD, at all Starlink sites). As a last resort, the problem may be regarded as a software bug and reported to the Starlink Software support mailing list (starlink@jiscmail.ac.uk).

10.2 BEGPLOT or DEVICE does not work in ICL

If the error message:

```
TOOFEWPARS   Not enough procedure parameters
```

is returned after the BEGPLOT or ENDPLOT commands have been invoked, the reason is that another application package or ICL procedure has switched parameter checking on within ICL (SG/5). The problem is simply overcome by typing the command:

```
ICL> set nocheckpars
```

The PONGO commands will then work correctly.

10.3 READF cannot find a column label

If a column label is used with the READF command instead of the column number (see §5) and READF fails, *e.g.* :

```
PONGO> readf myfile.dat xcol=1 ycol=3 zcol="ZCOL"
File: /disk/scratch/jbloggs/plots/myfile.dat
XCOL - 1 is column number 1.
YCOL - 3 is column number 3.
!! ZCOL incorrectly specified as ZCOL.
!  READF: Data could not be read.
```

then either the data file does not have column labels or the incorrect column label has been used. It is of particular significance here that the column labels are *case-sensitive*. In fact, JBLOGGS found that the column label for his ZCOL data was Zcol:

```
PONGO> readf myfile.dat xcol=1 ycol=3 zcol="Zcol"
File: /disk/scratch/jbloggs/plots/myfile.dat
XCOL - 1 is column number 1.
YCOL - 3 is column number 3.
ZCOL - Zcol is column number 5.
EXCOL - 0 is column number 0.
EYCOL - 0 is column number 0.
LABCOL - 0 is column number 0.
SYMCOL - 0 is column number 0.
42 data points read.
PONGO>
```

10.4 Strange behaviour of DLIMITS

If PONGO does not seem to be doing something correctly, it is often because there are unwanted data in the error columns. This can come about when a file that does not contain errors is read (the appropriate parameters to READF have been set to zero), following one where there have been errors. Here, it should be noted that setting a column parameter in READF to zero means that no data will be read from the file into that particular area, it does not clear the data values.

This is a feature rather than a bug, because it allows X and Y data from two different files to be plotted together, as long as each file contains the same number of points (the number of points read from the second file is assumed to be the number required). Although advantageous, it is possible for the internal data areas to get into a mess with this arrangement. If this seems to have happened then the command:

```
PONGO> clear data
```

should be used to clear all the data areas.

10.5 Error messages from SHOWPONGO in ICL

The commands SHOWPONGO may sometimes deliver the error messages

```
!! Object 'PONGO_<param>' not found.
!! DAT_FIND: Error finding a named component in an HDS structure.
```

when invoked (<param> refers to a PONGO parameter name in this context). The reason is simply that no global parameter of that name currently exists. This can either be because PONGO has not been used before (or not a lot), or because the ADAM global parameter file ($HOME/adam/GLOBAL.sdf) has been deleted since the last PONGO session. Although the error messages look alarming, they are harmless. To stop these error messages being output, the command RESETPONGO should be executed. This command will assign a default value to each of the PONGO global parameters except PONGO_DATA, the data file name used by READF, and SHOWPONGO will be subsequently more informative.

10.6 AGI problems

The AGI database is normally kept in a separate HDS file (SG/4) for each machine in your $HOME directory. The command BEGPLOT opens the AGI database and reads database information relevant to the graphics device being used. During a PONGO plotting session, the database file is held open to update the database as the plotting session proceeds. At the end of plotting, when ENDPLOT is executed, the update of the database is completed and the database file is closed. If another non-PONGO application which uses AGI is run before the current PONGO plotting session has been ended (*i.e.* using ENDPLOT), it will be unable to access the AGI database and will subsequently fail. In the event of this happening, using the PONGO ENDPLOT command will restore the correct behaviour of the non-PONGO application.

What is more important, in order to use AGI successfully, is the use of ENDPLOT before exiting ICL or CL. Because PONGO keeps the AGI database file open throughout a plotting session it is possible to exit ICL or CL without having first executed ENDPLOT. If this is done, it is likely that the AGI database will not have been fully updated before being closed, with the result that the next time AGI is used it will behave inconsistently. PONGO crashing during a plotting session (hopefully, a rare event) can also result in a corrupted AGI database. There is no solution to this problem other than to delete the database file and start again. The AGI database files are kept in the directory $HOME and have names of the form agi_<machine>.sdf, where <machine> is the name of the machine on which AGI is being used. This file may be deleted at any time before a BEGPLOT command or after an ENDPLOT command. The BEGPLOT command may then be used to begin a new PONGO plot, creating a new AGI database as a result.

10.7 RESET peculiarity (ICL only)

When using the ADAM parameter system (SG/4) `RESET` qualifier to set parameters back to their default values (this facility would be particularly useful for `BOXFRAME`, for example), there will be no effect on parameters which get their values from global parameters. Use the `RESETPONGO` command to reset global parameters, or alternatively (somewhat drastically) delete the `GLOBAL.sdf` file.

10.8 Plotting large numbers of positions

PONGO can store 5000 positions at any time. This can very occasionally cause problems for specific types of use. One way in which you can actually plot more positions than this, is by reading in the data in chunks (of 5000 points) and plotting these. The following ICL procedure shows one way in which you might do this:

```
proc superplot file, x1, x2, y1, y2, n
  begplot xwindows xmin=(x1) xmax=(x2) ymin=(y1) ymax=(y2)
  limits (x1) (x2) (y1) (y2)
  boxframe
  loop for i = 0 to (n) step 1
    j = i + 1
    print "Plotting points from" (i*5000) " to " (j*5000-1)
    readf data=(file) xcol=1 ycol=2 from=(i*5000) to=(j*5000-1) all accept
    points
  end loop
  endplot
end proc
```

This would then be invoked as in (assuming the procedure was kept in a file `superplot.icl`):

```
ICL > load superplot.icl
ICL > superplot mygalaxies.dat -3 3 -3 3 10
```

Which would plot 50000 positions stored in the file `mygalaxies.dat`.

11 References

Bailey, J.A. & Chipperfield, A.J., SG/5: ICL – The Interactive Command Language for ADAM.
Currie, M.J., SUN/95: KAPPA – Kernel Application Package.
Eaton, N., SUN/48: AGI – Applications Graphics Interface.
Giddings, J. & Rees, P., SUN/37: IUEDR – IUE Data Reduction package.
Lawden, M.D. & Hartley, K.F., SG/4: ADAM – The Starlink Software Environment.
Terrett, D.L., SUN/15: PGPLOT – Graphics Subroutine Library.
Pearson, T.J., MUD/061: PGPLOT – Graphics Subroutine Library.

A Alphabetical List of Commands

In the following list the parts of the command names outside parentheses define the minimum abbreviation for that application (note that IRAF/CL applies a different set of abbreviations).

ADV(ANCE) – Clear the graphics screen.

ANN(OTATE) – Annotate the plotted data.

ARC – Draw an arc of an ellipse.

AVEDAT – Average the data in the XCOL and YCOL areas.

BEGP(LOT) – Open a plotting device.

BIN – Plot a histogram using previously binned data.

BOX(FRAME) – Draw a frame and axes on the plot.

CCMATH – Perform inter-column maths.

CHANGE – Change plotting attributes.

CLEAR – Clear plotting attributes.

CLOG – Take the logarithm of a column.

CONN(ECT) – Draw straight lines between the data points.

CURSE – Use the cursor to get interactive input.

DATA – Specify the data file name.

DEGTOR – Convert the specified data area from degrees to radians.

DEVICE – Open a plotting device.

DLIM(ITS) – Set the world coordinate limits using the data range.

DRAW – Draw a line from the current pen position the specified point.

DRAWPOLY – Draw a polygon.

ELLIPSES – Draw error ellipses.

ENDP(LOT) – Close down the current plotting device.

ERASE – Clear the graphics screen.

ERRORBAR – Draw error bars on the plotted data.

ERRX – Draw symmetric error bars in the X direction.

ERRY – Draw symmetric error bars in the Y direction.

EXC(OLUMN) – Specify the column containing the X-axis error data.

EXPA(ND) – Set the character height.

EYC(OLUMN) – Specify the column containing the Y-axis error data.

FILLSTY – Change fill-style plotting attributes.

FITC(URVE) – Fit a curve to the data.

FITL(INE) – Fit a straight line to the data.

FONT – Set the text font.

GETP(OINT) – Retrieve information for a plotted data point.

GPOINTS – Plot points or lines between the data.

GRID – Draw a coordinate grid at specified intervals.

GT_CIRCLE – Draw a great circle between two points.

HIST(OGRAM) – Bin and plot a histogram of the data.

INQ(UIRE) – Display PONGO status information.

LABC(OLUMN) – Specify the column used for data labels.

LAB(EL) – Draw the axis labels and title on the plot.

LIM(ITS) – Set the world coordinate limits.

LT(YPE) – Set the line style.

LWE(IGHT) – Set the line width.

MARK – Draw a point mark at the specified position.

MOVE – Set the current pen position.

MTEXT – Draw a text string relative to the viewport.

PALET(TE) – Change the plotting pen colours.

PAPER – Change the size and aspect ratio of the plotting surface.

PCOL(UMN) – Specify the column used for symbol codes.

PEN – Set the current pen.

PLOTF(UN) – Plot a given function.

PLOTH(IST) – Plot a histogram of the data.

POI(NTS) – Draw a point mark at each of the data points.

PRIM – Perform primitive plotting functions.

PTEXT – Draw a text string at the specified position and angle.

PTINFO – Get the coordinates of a specified data point.

PVECT – Draw proper motion vectors.

RADIATE – Draw a line from the given point to the first NP data points.

READ(F) – Read from a formatted data file.

RESETP(ONGO) – Reset the state of PONGO.

RTODEG – Convert the specified data area from radians to degrees.

SETPROJ – Set the projection geometry.

SHOWP(ONGO) – Show the PONGO global parameters.

SIZE(PLOT) – Draw point marks of differing sizes at each of the data points.

SYMC(OLUMN) – Specify the column used for symbol codes.

TEXT – Draw a text string on the plot at the specified position.

VECT – Draw vectors from each data point.

VIEWPORT – Set the viewport for the current plotting device.

VPORT – Set the viewport using normalised device coordinates.

VP_BH – Set the viewport to the bottom half of the plotting surface.

VP_BL – Set the viewport to the bottom-left quarter of the plotting surface.

VP_BR – Set the viewport to the bottom-right quarter of the plotting surface.

VP_TH – Set the viewport to the top half of the plotting surface.

VP_TL – Set the viewport to the top-left quarter of the plotting surface.

VP_TR – Set the viewport to the top-right quarter of the plotting surface.

VSIZE – Set the viewport using its physical size in inches.

VSTAND – Set the standard viewport.

WNAD – Adjust the viewport so that the X and Y scales are the same.

WORLD – Set the world coordinates for the plot.

WRITE(I) – Write information to an output file.

WTEXT – Draw a text string on the plot.

XC(OLUMN) – Specify the column containing the X-axis data.

XERR – Draw symmetric error bars in the X direction.

XLIN(EAR) – Put 1. . . N into the XCOL data area.

XLOG(ARITHM) – Take the logarithm of the X-axis data.

XOFF(SET) – Add a constant offset to the X-axis data.

XSCALE – Multiply the values in the XCOL and EXCOL data areas by a constant.

YC(OLUMN) – Specify the column containing the Y-axis data.

YERR – Draw symmetric error bars in the Y direction.

YLIN(EAR) – Put 1. . . N into the YCOL data area.

YLOG(ARITHM) – Take the logarithm of the Y-axis data.

YOFF(SET) – Add a constant offset to the Y-axis data.

YSCALE – Multiply the values in the YCOL and EYCOL data areas by a constant.

ZC(OLUMN) – Specify the column containing the Z-axis data.

ZSCALE – Multiply the values in the ZCOL data area by a constant.

B PONGO Command Definitions

This section gives detailed descriptions for each of the PONGO commands. The commands are listed in alphabetical order. A description of the command is followed by a usage section and then a description of each of the command parameters and its action. The usage section shows the order of any positional parameters and lists any other required values. Positional parameters may be optional, if a default exists, in which case they are shown in []. For example the usage section of `BEGPLOT` looks like:

```
Usage: begplot [device] [action] [clear] [overlay]
           { label=?
       action
```

In this case all the positional parameters ([device] [action] [clear] and [overlay]) are optional and have defaults. This example also shows that the use of a non-positional parameter 'label' is dependent on the value of the parameter 'action' (in fact it will only be used if action is set to 'L'). You need to consult the full parameter descriptions to appreciate this. Non-positional parameters that are normally defaulted are not shown in the usage section.

The full parameter descriptions include default behaviour; *i.e.* what value is taken if a parameter value is not given on the command line. For simple cases, this behaviour is described in **[** *square brackets* **]**; for more complicated behaviour, a full description is provided. An empty set of square brackets, *i.e.* **[]**, indicates that a parameter value *must* be specified on the command line.

Commands that are described as ICL procedures can only take parameters on the command-line. These should be given as single values in the correct order – not as `keyword=value` pairs.

Commands that are described as synonyms such as CHANGE LINEWID= must be given a single value for the first parameter.

If you are using CL then still consult this section, but use the on-line help and the `lparam` command to determine the order and availability of any parameters. You should also ignore any references to global parameters (and in particular the behaviour of parameters when they are specified on the command-line).

ADVANCE
Clear the graphics screen

Description:

The plotting surface is cleared.

This command is a synonym for CLEAR SCREEN.

Usage:

```
advance
```

ANNOTATE
Annotate the plotted data

Description:

Each of the points on the plot is labelled with the appropriate internal label (if it has been read from the data file). If no parameters are specified, the default action is for the label to be written with a zero offset in X and an offset of approximately one character height in Y.

Usage:

```
annotate [xoff] [yoff] [justification]
```

Parameters:

XOFF = _REAL (Read and Write)

The X coordinate offset of the string relative to each data point. The application will use the value 0.0 (*i.e.* no offset) unless a value is given on the command line. [0.0]

YOFF = _REAL (Read and Write)

The Y coordinate offset of the string relative to each data point.

The application will prompt with a value of about 1/40th of the height of the viewport unless a value is given on the command line.

[1/40th of the viewport height.]

JUSTIFICATION = _REAL (Read and Write)

The justification about the point specified by XOFF and YOFF relative to each data point (in the range 0.0 to 1.0). Here, 0.0 means left justify the text relative to the data point, 1.0 means right justify the text relative to the data point, 0.5 means centre the string on the data point, other values will give intermediate justifications.

If no value is specified on the command line, the current value is used. The current value is initially set to 0.0.

PROJECTION = _CHAR (Read)

The projection that has been used to plot the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR", and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

ARC
Draw an arc of an ellipse

Description:

A specified arc of an ellipse is drawn from the position angles of the start and end of the arc, the semi axes, the position of the centre and the rotation of the axes. If no parameters are specified then whole ellipses can be drawn from the data stored in the following data areas:

- XCOL – X centre,
- YCOL – Y centre,
- EXCOL – semi-major axis,
- EYCOL – semi-minor axis.

Usage:

```
arc [a] [b] [x0] [y0] [pastart] [paend] [rotation]
```

Parameters:

A = _REAL (Read and Write)

The semi-major axis of the ellipse.

If no value is specified on the command line, the current value is used. If there is no current value, a default value 1.0 is used.

B = _REAL (Read and Write)

The semi-minor axis of the ellipse.

If no value is specified on the command line, the current value is used. If there is no current value, a default value 1.0 is used.

X0 = _DOUBLE (Read and Write)

The X coordinate of the centre of the ellipse.

If no value is specified on the command line, the current value is used. If there is no current value, a default value 0.0 is used.

Y0 = _DOUBLE (Read and Write)

The Y coordinate of the centre of the ellipse.

If no value is specified on the command line, the current value is used. If there is no current value, a default value 0.0 is used.

PASTART = _REAL (Read and Write)

The position angle of the start of the arc (degrees).

If no value is specified on the command line, the current value is used. If there is no current value, a default value 0.0 is used.

PAEND = _REAL (Read and Write)

The position angle of the end of the arc (degrees).

If no value is specified on the command line, the current value is used. If there is no current value, a default value 360.0 is used.

ROTATION = _REAL (Read and Write)

The angle that the major axis makes with the horizontal (degrees anti-clockwise).

If no value is specified on the command line, the current value is used. If there is no current value, a default value 0.0 is used.

FROMDATA = _LOGICAL (Read)

If TRUE, the command will use the data already loaded to draw whole ellipses, with positions and sizes specified as above. [FALSE]

PROJECTION = _CHAR (Read)

The geometry that is to be used to plot the arc. This is explained in detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. If no value is specified on the command line, the current value is used. If there is no current value, a default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

ERSCALE = _REAL (Read and Write)

The factor used to scale values in the EXCOL and EYCOL data areas. This allows the ellipse axes lengths to be scaled, changing the sizes of ellipses produced using the FROMFILE parameter.

This parameter is not specified on the command line. The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.

Examples:

```
PONGO> ARC 1 1 0 0
```

will draw a circle of radius 1 (world coordinates), assuming the PASTART and PAEND parameters have their default values (0.0 and 360.0 degrees respectively).

```
PONGO> ARC FROMDATA
```

Draws arcs using the information read into the XCOL YCOL EXCOL and EYCOL data areas.

AVEDAT
Average the data in the XCOL and YCOL areas

Description:

Re-bin the XCOL and YCOL data, averaging the data in each sample, and puts the result back into the XCOL and YCOL areas. The standard deviations of the averages are put into the EXCOL and EYCOL areas. There are two ways in which the averaging may be done:

- the data may be split into N equally sized bins over the X range, and the values in each bin averaged;
- the data may be averaged in groups of N data.

Usage:

```
avedat action [nbin]
```

Parameters:

ACTION = _CHAR (Read)

The type of binning be used for the averaging. If "X", the data are divided into NBIN bins over the X range. If "N", bins of varying widths with each containing NBIN data points are formed.

[The value will be prompted for. It has the default "X".]

NBIN = _INTEGER (Read)

Depending upon the value of ACTION, either the number of bins (ACTION="X"), or the number of points per bin (ACTION="N").

[The value will be prompted for. It has the default 10.]

XMIN = _REAL (Read)

The minimum X value to be used in the average.

[The value of the global parameter PONGO_XMIN is used. If PONGO_XMIN is not defined, the default value 0.0 is used.]

XMAX = _REAL (Read)

The maximum X value to be used in the average.

[The value of the global parameter PONGO_XMAX is used. If PONGO_XMAX is not defined, the default value 1.0 is used.]

BEGPLOT
Open a plotting device

Description:

Set up a device for subsequent PONGO plotting commands. This application allows plotting onto an AGI picture created by a different package (*e.g.* KAPPA), or the creation of a new base picture.

If a picture created by a run of a previous application is made (using either the "current" picture or selecting using a known picture label see parameter ACTION) then it is possible to overlay new graphics (such as annotations, lines *etc.*) using the same coordinate system (see the OVERLAY parameter).

This command is an ICL hidden procedure which uses the undocumented PONGO application BEGPONGO.

Usage:

```
begplot [device] [action] [clear] [overlay]
          {  label=?
   action
```

Parameters:

ACTION = _CHAR (Read and Write)

If 'B', the plotting device will be cleared and the whole of its plotting surface used. If equal to 'C', the current picture will be used and a PGPLOT viewport created inside it. If 'L' then a previously labelled picture (set using KAPPA:PICLABEL) can be selected. Once set, this parameter will retain its value in subsequent invocations of BEGPLOT. ['C']

CHEIGHT = _REAL (Write)

The character height scaling factor. A value of 1.0 implies a nominal character height of 1/40th the viewport height. The value is set by the application from the height of the chosen picture (unless the picture is the base one). The result is written to the global parameter PONGO_CHEIGHT.

CLEAR = _LOGICAL (Read)

If TRUE then the current picture will be cleared of any existing graphics. [TRUE]

DEVICE = DEVICE (Read and Write)

The name of the device to be used for plotting. The names of the currently available devices can be found using the `INQUIRE DEVICES=YES` command.

The value of the global parameter GRAPHICS_DEVICE is used unless a value is specified on the command line. If GRAPHICS_DEVICE is not defined and no value is specified on the command line, the value will be prompted for.

LABEL = _CHAR (Read)

If ACTION=L is selected then the name of the AGI picture to be selected is given by this parameter. AGI pictures can be labelled using the KAPPA application PICLABEL. ['']

OVERLAY = _LOGICAL (Read)

If TRUE, the PGPLOT viewport created will exactly overlay the the last DATA picture. This is useful for drawing axis labels using `BOXFRAME` on an image or contour map *etc.* that has been displayed by another package (*e.g.* KAPPA:DISPLAY). [FALSE]

XMIN = _REAL (Write)

The left hand edge of the world coordinate limits from the selected picture. This defaults to 0.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_XMIN.

XMAX = _REAL (Write)
The right hand edge of the world coordinate limits from the selected picture. This defaults to 1.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_XMAX.

YMIN = _REAL (Write)
The lower edge of the world coordinate limits from the selected picture. This defaults to 0.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_YMIN.

YMAX = _REAL (Write)
The lower edge of the world coordinate limits from the selected picture. This defaults to 1.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_YMAX.

BIN
Plot a histogram using previously binned data

Description:

The data in the XCOL and YCOL data areas are used to plot a histogram. The data in the XCOL area specify the bin edges, while the data in the YCOL area specify the respective frequency for each bin.

This command is a synonym for `PLOTHIST B`.

Usage:

```
bin
```

Parameters:

CENTRE = _LOGICAL (Read)

This parameter specifies whether the values in the XCOL data area denote the centre of each bin (when TRUE) or its lower edge (when FALSE). [FALSE]

BOXFRAME
Draw a frame and axes on the plot

Description:

Draw a frame, axes and tick-marks on a plot. The application has great flexibility in the type of axis labelling that can be produced – it is essentially an interface to the PGPLOT routines PGBOX and PGTBOX.

Usage:

```
boxframe [xopt] [yopt] [xtic] [ytic] [nxsub] [nysub]
```

Parameters:

XOPT = _CHAR (Read and Write)

A string that controls the style of the X-axis labelling and tick-marks. It consists of a series of letters, which have the meanings shown below (reproduced from the PGPLOT manual):

- "A" – Draw an axis (the X axis is the horizontal line Y=0, the Y axis is the vertical line X=0).
- "B" – Draw the bottom (X) or left (Y) edge of the frame.
- "C" – Draw the top (X) or right (Y) edge of the frame.
- "G" – Draw a grid of vertical (X) or horizontal (Y) lines.
- "I" – Invert the tick-marks (*i.e.* draw them outside the viewport instead of inside).
- "L" – Label the axis logarithmically (see below).
- "N" – Write numeric labels in the conventional location below the viewport (X) or to the left of the viewport (Y).
- "P" – Extend (project) major tick-marks outside the box (ignored if option I is specified).
- "M" – Write numeric labels in the unconventional location above the viewport (X) or to the right of the viewport (Y).
- "T" – Draw major tick-marks at the major coordinate interval.
- "S" – Draw minor tick-marks (sub-ticks).
- "V" – Orient numeric labels vertically (this is only applicable to Y – the default is to write Y-axis labels parallel to the axis).
- "1" – Force decimal labelling.
- "2" – Force exponential labelling.

A set of special letters control the plotting of RA and DEC axes.

- "Z" for (DD) HH MM SS.S time labelling
- "H" means superscript numbers with d, h, m, & s symbols
- "D" means superscript numbers with o, ', & '' symbols
- "F" causes the first label (left- or bottom-most) to be omitted. Useful for sub-panels that abut each other. Care is needed because first label carries sign as well.
- "O" means omit leading zeros in numbers < 10 *e.g.* 3h 3m 1.2s rather than 03h 03m 01.2s Useful to help save space on X-axes. The day field does not use this facility.

Note that to use these features your data values should be in radians (PONGO automatically converts HH:MM:SS.SSS strings to radians, and the DEGTOR command from degrees to radians).

[The global parameter PONGO_XOPT is used. If PONGO_XOPT is not defined, the default value "BCNST" is used.]

YOPT = _CHAR (Read and Write)

A string that controls the style of the Y-axis labelling and tick-marks. It consists of a series of letters, as given for the parameter XOPT.

[The global parameter PONGO_YOPT is used. If PONGO_YOPT is not defined, the default value "BCNST" is used.]

XTIC = _REAL (Read and Write)

The major tick-mark interval on the X-axis. If XTIC is set to 0.0, PGPLOT makes a sensible choice.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

YTIC = _REAL (Read and Write)

The major tick-mark interval on the Y-axis. If YTIC is set to 0.0, PGPLOT makes a sensible choice.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

NXSUB = _INTEGER (Read and Write)

The number of minor tick-marks between each major tick-mark on the X-axis. If NXSUB is set to 0, PGPLOT makes a sensible choice.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.

NYSUB = _INTEGER (Read and Write)

The number of minor tick-marks between each major tick-mark on the Y-axis. If NYSUB is set to 0, PGPLOT makes a sensible choice.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.

Notes:

- Right Ascension and Declination axes.

It is possible to draw RA and DEC axes using this routine. To do this read in your X and Y data in radians (this is the default if your data are stored in the HH:MM:SS.SSS, DD:MM:SS.SSS formats) and then set the "Z" character in the XOPT and YOPT parameters. A good combination of options are:

```
XOPT='BCNSTZHG' YOPT='BCNSTZDG'
```

For more about this (in particular how the interval options are set) see the PGPLOT documentation about PGTBOX. Note that the "Y" option is always set and that the Y axis values are converted into pseudo arc seconds.

CCMATH
Perform inter-column maths

Description:

Perform inter-column maths (using TRANSFORM, SUN/61) The expressions and functions recognised have Fortran types and syntax. Any construct that is legal in TRANSFORM is legal in this subroutine, with the additional function INDEX for filling the array with an increasing sequence of integers. See SUN/61 for further details.

The names used for the data areas are as follows:

- "X" – the XCOL data area,
- "Y" – the YCOL data area,
- "Z" – the ZCOL data area,
- "EX" – the EXCOL error area,
- "EY" – the EYCOL error area.

Usage:

```
ccmath
```

Parameters:

X = _CHAR (Read)

The transformation to perform on the contents of the XCOL data area.

["X – *i.e.* will cause the contents of the data area to remain unchanged.]

Y = _CHAR (Read)

The transformation to perform on contents of the YCOL data area.

["Y" – *i.e.* will cause the contents of the data area to remain unchanged.]

Z = _CHAR (Read)

The transformation to perform on contents of the ZCOL data area.

["Z" – *i.e.* will cause the contents of the data area to remain unchanged.]

EX = _CHAR (Read)

The transformation to perform on contents of the EXCOL data area.

["EX" – *i.e.* will cause the contents of the data area to remain unchanged.]

EY = _CHAR (Read)

The transformation to perform on contents of the EYCOL data area.

["EY" – *i.e.* will cause the contents of the data area to remain unchanged.]

Examples:

```
PONGO> CCMATH X=2*Y
```

will fill each element of the XCOL data area with twice the corresponding element of the YCOL data area.

Notes:

- More than one array may be manipulated with a single command.
- The INDEX function cannot be combined with any other function.

CHANGE
Change plotting attributes

Description:

Change the PGPLOT plotting attributes: *e.g.* line style, pen colour *etc.* Several of the attributes can be changed at the same time. Each of the parameters is remembered from the last invocation of `CHANGE`: after `BEGPLOT` has been run, a single invocation of `CHANGE` can be used to reset the plotting attributes to their values the last time PONGO was used.

Usage:

```
change
```

Parameters:

ANGLE = _REAL (Read)

The angle hatch lines make with horizontal, in degrees. Only used when FILLSTY is 3. [45.0]

COLOUR = _INTEGER (Read and Write)

The pen number (colour index) PGPLOT uses for plotting. The value should be between 0 and 255. Usually the first 16 pens are predefined to have the following colours:

- 0 – background,
- 1 – foreground (default),
- 2 – red,
- 3 – green,
- 4 – blue,
- 5 – cyan,
- 6 – magenta,
- 7 – yellow,
- 8 – red + yellow (orange),
- 9 – green + yellow,
- 10 – green + cyan,
- 11 – blue + cyan,
- 12 – blue + magenta,
- 13 – red + magenta,
- 14 – dark grey,
- 15 – light grey.

It is possible to change the colour representation of any of the pen colour indices using the PALETTE application.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1 (*i.e.* foreground). [1]

CHEIGHT = _REAL (Read and Write)

The character height scaling. This parameter scales the default character height and also alters the size of the tick marks and symbols that PGPLOT plots. The default character height in PGPLOT is about 1/40 of the viewport height.

[The value of the global parameter PONGO_CHEIGHT is used. If PONGO_CHEIGHT is not defined, the default value 1.0 is used.]

FONT = _INTEGER (Read)

The font used by PGPLOT. The styles for each font are as follows:

- 1 – single-stroke font (default),
- 2 – roman font,
- 3 – italic font,
- 4 – script font.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1 (*i.e.* single-stroke font). [1]

FILLSTY = _INTEGER (Read)

The fill style used by PGPLOT. The fill styles are as follows:

- 1 – solid fill,
- 2 – hollow fill,
- 3 – hatched,
- 4 – cross-hatched

If the value is not specified on the command line, the current value is used. The current value is initially set to 2 (*i.e.* hollow fill). [2]

LINESTY = _INTEGER (Read)

The line style used by PGPLOT. The line style may be one of the following:

- 1 – full line (default),
- 2 – dashed,
- 3 – dot-dash-dot-dash,
- 4 – dotted,
- 5 – dash-dot-dot-dot.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1 (*i.e.* full line). [1]

LINEWID = _INTEGER (Read)

The line width scaling. This parameter scales the default line width.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1. [1]

PHASE = _REAL (Read)

The fraction of SEPARATION that hatched lines are displaced. Modifying this between regions makes their separation more obvious. This is only used when FILLSTY is 3. [0]

SEPARATION = _REAL (Read)

The separation of hatched lines. The unit spacing is 1 percent of the smaller of the height or width of the view surface. Only used when FILLSTY is 3. [1]

TEXTBACK = _INTEGER (Read)

The pen number (colour index) of the background used when drawing text. If less than zero then a transparent background is used. Zero erases the plot under the text region. [-1]

Examples:

```
ICL> CHANGE RESET
```

will reset the plotting attributes to their default values.

CLEAR
Clear plotting attributes

Description:

Clear or reset various PONGO plotting attributes. Several or all of these plotting attributes can be specified on the command line.

Usage:

```
clear
```

Parameters:

SCREEN = _LOGICAL (Read)

If TRUE, the plotting surface will be cleared. [FALSE]

DATA = _LOGICAL (Read)

If TRUE, the data arrays will be cleared. This can be useful if there is still unwanted data left in some of the data areas: for example, the `READF` command does not automatically clear columns that it does not read in. Even if the appropriate `READF` parameters have been set to 0 to inhibit reading error columns from the second file, the values from the first file will remain in the data areas. This behaviour is designed to allow columns to be read from separate files and combined in one plot. However, it can cause the automatic axis limit finding routines to appear to fail because of data left in error columns. This most commonly occurs when a file not containing error values is read immediately after one that does. [FALSE]

LIMITS = _LOGICAL (Read)

If TRUE, the data limits will be cleared and the data re-examined in order to determine the axis limits. This command is often necessary if a complicated function has been performed on the data using the `CCMATH` command, because the current PONGO parameters may no longer be applicable to the data. [FALSE]

LABLST = _LOGICAL (Read)

If TRUE, the internal label list generated by the CURSE command will be cleared. [FALSE]

AGI = _LOGICAL (Read)

If TRUE, the AGI database is cleared for the current device. [FALSE]

ERSCALE = _REAL (Read)

The scale factor by which the errors are to be multiplied. (Can only be set when LIMITS are cleared.)

The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.

CLOG
Take the logarithm of a column

Description:

Take the base 10 logarithm of a data column. This application should be used to take the logarithm of the data columns in preference to doing it with `CCMATH`, because it can deal with the associated error values consistently. It automatically adds the "L" option to the PONGO_XOPT or PONGO_YOPT global parameters, as appropriate.

Usage:

```
clog action
```

Parameters:

ACTION = _CHAR (Read)

The data column to transform. It should be one of the following:

- "X" – XCOL,
- "Y" – YCOL,
- "Z" – ZCOL.

[The value will be prompted for.]

XOPT = _CHAR (Write)

The PGPLOT X-axis options string. The global parameter will be updated to include the PGPLOT "L" axis option at the start. This option means that logarithmic style axis labels and tick marks will be plotted. The `READF` command will automatically remove any "L" characters at the start of this string since it assumes that they have been put there by CLOG, and fresh data read in will not be logarithmic. If data are naturally logarithmic, the "L" should be placed other than at the start of the string to make an "L" option that will not be modified by PONGO.

It is not intended that this parameter be set by you when CLOG is executed.

The value will be written to the global parameter PONGO_XOPT.

YOPT = _CHAR (Write)

The PGPLOT Y-axis options string. Its action is similar to the XOPT parameter. It is not intended that this parameter be set by you when CLOG is executed.

The value will be written to the global parameter PONGO_YOPT.

CONNECT
Draw straight lines between the data points

Description:

Straight line segments are drawn between the data points in the `XCOL` and `XCOL` data areas. This may not be used with projections other than `"NONE"`

This command is a synonym for `GPOINTS C`.

Usage:

```
connect
```

CURSE
Use the cursor to get interactive input

Description:

Display the cursor and perform various interactive tasks. These tasks are as follows:

- Q – QUIT The application ends.
- D – DRAW A line is drawn from the PGPLOT "current position" to the cursor position. This position is also written to the internal label list.
- E – EXPAND The plotting limits are expanded by a factor of 2 about the position of the cursor. No other action is taken. This allows the screen to be cleared and the graph to be re-plotted without having to set the limits explicitly.
- G – GRADIENT Draw a line between two consecutive cursor hits and report the gradient of the line.
- L – LABEL Write a label directly onto the plot. The application uses up to two points which define the angle at which the label is to be drawn. Once "L" has been pressed, you are given the option to add another point in the standard PGPLOT fashion (c.f. the PGPLOT routine PGNCURSE); *i.e.* :

  - A – add a point
  - D – delete a point
  - X – finish

  On pressing the "X" key, a label is prompted for. If only one point is supplied, the label is plotted horizontally.
- M – MARK Mark the current cursor position with the current symbol type.
- O – ANNOTATE The label for the nearest data point is written with its right hand end at the position defined by the cursor. The information to create this label is stored in an internal table. This information can be written out into file suitable for including in an ICL command file by using the WRITEI application, *e.g.* :
  PONGO> WRITEI LABLST
- P – ANNOTATE The label for the nearest data point is written with its left hand end at the position defined by the cursor. The information to create this label is stored in an internal table. This information can be written out into file suitable for including in an ICL command file by using the WRITEI application, *e.g.* :
  PONGO> WRITEI LABLST
- S – SHRINK The plotting limits are set so as to zoom out by a factor of 2 about the position of the cursor. No other action is taken. This allows the screen to be cleared and the graph to be re-plotted without having to set the limits explicitly.
- V – MOVE The PGPLOT "current position" will be set to the cursor position. This position is also written to the internal label list.
- X – POSITION The current position of the cursor in world coordinates is written to the terminal and the XCURSE and YCURSE parameters.
- Z – ZOOM The limits for a zoomed version of the current plot are set. The application will prompt for two points which define the bottom left corner and the top right corner of the new graph surface.

- A – add a point
- D – delete a point
- X – finish

Usage:

```
curse [symbol] label=?
```

Parameters:

SYMBOL = _INTEGER (Read)

The symbol number used in the MARK option.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.

LABEL = _CHAR (Read)

The label to be written to the screen with the LABEL option.

[The value is prompted for.]

JUSTIFICATION = _REAL (Read and Write)

The justification about the point (in the range 0.0 to 1.0). Here, 0.0 means left justify the text relative to the data point, 1.0 means right justify the text relative to the data point, 0.5 means centre the string on the data point, other values will give intermediate justifications.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.5 (*i.e.* centre the text).

XCURSE = _REAL (Write)

The X-axis position of the last graphics cursor hit when using the "X" option.

The value is written to the PONGO_XCURSE global parameter.

YCURSE = _REAL (Write)

The Y-axis position of the last graphics cursor hit when using the "X" option.

The value is written to the PONGO_YCURSE global parameter.

XMIN = _REAL (Write)

The left hand edge of the world coordinate limits. The value is set by the application in the zooming options. It is not intended that the value be specified on the command line.

The value is written to the global parameter PONGO_XMIN.

XMAX = _REAL (Write)

The right hand edge of the world coordinate limits. The value is set by the application in the zooming options. It is not intended that the value be specified on the command line.

The value is written to the global parameter PONGO_XMAX.

YMIN = _REAL (Write)

The lower edge of the world coordinate limits. The value is set by the application in the zooming options. It is not intended that the value be specified on the command line.

The value is written to the global parameter PONGO_YMIN.

YMAX = _REAL (Write)

The upper edge of the world coordinate limits. The value is set by the application in the zooming options. It is not intended that the value be specified on the command line.

The value is written to the global parameter PONGO_YMAX.

PROJECTION = _CHAR (Read)

The projection that has been used to plot the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

DATA
Specify the data file name

Description:

The file name for the input data is set.

This command is a synonym for SETGLOBAL PONGO_DATA.

Usage:

```
data filename
```

Parameters:

DATA = FILENAME (Write)

The name of the formatted data file. []

DEGTOR
Convert the specified data area from degrees to radians

Description:

The values in the specified data area are converted from degrees to radians.

This command is an ICL hidden procedure which uses the `CCMATH` application.

Usage:

```
degtor column
```

Parameters:

COLUMN = _CHAR (Read and Write)

The column to be converted from degrees to radians. []

DEVICE
Open a plotting device

Description:

Set up a device for subsequent PONGO plotting commands. This application allows plotting onto an AGI picture created by a different package (*e.g.* KAPPA), or the creation of a new base picture.

If a picture created by a run of a previous application is made (using either the "current" picture or selecting using a known picture label see parameter ACTION) then it is possible to overlay new graphics (such as annotations, lines *etc.*) using the same coordinate system (see the OVERLAY parameter).

This command is a synonym for `BEGPLOT`.

Usage:

```
device [device] [action] [clear] [overlay]
    { label=?
  action
```

Parameters:

ACTION = _CHAR (Read and Write)

If `’B’`, the plotting device will be cleared and the whole of its plotting surface used. If equal to `’C’`, the current picture will be used and a PGPLOT viewport created inside it. If `’L’` then a previously labelled picture (set using KAPPA:PICLABEL) can be selected. Once set, this parameter will retain its value in subsequent invocations of `BEGPLOT`. [’C’]

CHEIGHT = _REAL (Write)

The character height scaling factor. A value of 1.0 implies a nominal character height of 1/40th the viewport height. The value is set by the application from the height of the chosen picture (unless the picture is the base one). The result is written to the global parameter PONGO_CHEIGHT.

CLEAR = _LOGICAL (Read)

If TRUE then the current picture will be cleared of any existing graphics. [TRUE]

DEVICE = DEVICE (Read and Write)

The name of the device to be used for plotting. The names of the currently available devices can be found using the `INQUIRE DEVICES=YES` command.

The value of the global parameter GRAPHICS_DEVICE is used unless a value is specified on the command line. If GRAPHICS_DEVICE is not defined and no value is specified on the command line, the value will be prompted for.

LABEL = _CHAR (Read)

If ACTION=L is selected then the name of the AGI picture to be selected is given by this parameter. AGI pictures can be labelled using the KAPPA application PICLABEL. [’’]

OVERLAY = _LOGICAL (Read)

If TRUE, the PGPLOT viewport created will exactly overlay the the last DATA picture. This is useful for drawing axis labels using `BOXFRAME` on an image or contour map *etc.* that has been displayed by another package (*e.g.* KAPPA:DISPLAY). [FALSE]

XMIN = _REAL (Write)

The left hand edge of the world coordinate limits from the selected picture. This defaults to 0.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_XMIN.

XMAX = _REAL (Write)

The right hand edge of the world coordinate limits from the selected picture. This defaults to 1.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_XMAX.

YMIN = _REAL (Write)

The lower edge of the world coordinate limits from the selected picture. This defaults to 0.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_YMIN.

YMAX = _REAL (Write)

The lower edge of the world coordinate limits from the selected picture. This defaults to 1.0 for a new (*i.e.* not overlaid) picture. The result is written to the global parameter PONGO_YMAX.

DLIMITS
Set the world coordinate limits using the data range

Description:

The ranges of the data given in the XCOL and XCOL data areas are used to set the world coordinate limits for plotting. A small border is also added to the data limits when calculating the world coordinate limits.

This command is a synonym for `WORLD DATA`.

Usage:

```
dlimits
```

Parameters:

XMIN = _REAL (Read and Write)

The world coordinate of the left-hand edge of the plot.

The application will determine the value.

XMAX = _REAL (Read and Write)

The world coordinate of the right-hand edge of the plot.

The application will determine the value.

YMIN = _REAL (Read and Write)

The world coordinate of the lower edge of the plot.

The application will determine the value.

YMAX = _REAL (Read and Write)

The world coordinate of the upper edge of the plot.

The application will determine the value.

PROJECTION = _CHAR (Read and Write)

The geometry to be used for plotting the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

[The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.]

RACENTRE = _CHAR (Read and Write)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.]

DECCENTRE = _CHAR (Read and Write)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.]

ERSCALE = _REAL (Read)

The scale factor to be applied to the EXCOL and EYCOL data when determining the limits of the world co-ordinates. [The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.]

DRAW
Draw a line from the current pen position to the specified point

Description:

A straight line is drawn from the current pen position to the specified point.

This command is a synonym for PRIM D.

Usage:

```
draw x y
```

Parameters:

X = _REAL (Read and Write)

The X coordinate of the point.

If no value is specified on the command line, the value is prompted for.

Y = _REAL (Read and Write)

The Y coordinate of the point.

If no value is specified on the command line, the value is prompted for.

PROJECTION = _CHAR (Read)

The geometry that is to be used for plotting. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".This parameter is only relevant when marking points.

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

DRAWPOLY
Draw a polygon

Description:

This routine draws a polygon using the data in the XCOL and YCOL areas to define the vertices of the region. The polygon is shaded using the current fill-area style attributes (see the FILLSTY command).

Usage:

```
drawpoly [fill]
```

Parameters:

FILL = _LOGICAL (Read)

Whether to fill the polygon or not. If TRUE then the current fill-area style attributes as set by the CHANGE command will be used. [TRUE]

PROJECTION = _CHAR (Read)

The geometry to be used for plotting the polygon. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

[The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.]

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.]

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.]

Examples:

```
drawpoly
```

This draws a single polygon on the current graphics display. The polygon is filled or shaded using the current fill-hatch styles.

```
drawpoly projection=aitoff deccentre="12:00:00"
```

This draws a single polygon on the current graphics display. The polygon is filled or shaded using the current fill-hatch styles and the polygon is drawn using an AITOFF projection. The lines along the edges of the polygon are drawn as suitable arcs.

Notes:

- The XCOL and YCOL positions are assumed to be in radians.
- If a projection is used then expect the polygon to be incorrect if it crosses any discontinuities, or if the edges are severely distorted (the polygon is filled as plotted on the display surface, not the sky).

ELLIPSES
Draw error ellipses

Description:

Draw error ellipses at each of the data points using values in the EXCOL and EYCOL error data areas, and the ZCOL data values as the normalised covariance. Depending upon the value of the AXES parameter, the major and minor axes of the ellipses will also be drawn.

The size of the ellipses depends upon the parameter ERSCALE which should be set before the data are read in so that the WORLD application can calculate the viewing area properly. This allows the ellipse size to be varied so that it can be drawn for different confidence levels.

ERSCALE	Confidence level
1.00	46%
2.30	68.3%
4.61	90%
9.21	99%

Usage:

```
ellipses [axes] [erscale]
```

Parameters:

AXES = _LOGICAL (Read and Write)

If TRUE, the axes of the ellipses will be drawn.

If the value is not specified on the command line, the current value is used. The current value is initially set to FALSE.

ERSCALE = _REAL (Read and Write)

Scale the error ellipses as described above. If the command WORLD DATA (which automatically sets the graph limits) is to work properly, the global parameter PONGO_ERSCALE should have been set by READF. However, if this facility is not required, it is perfectly acceptable to set ERSCALE when invoking ELLIPSES.

[The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.]

ENDPLOT
Close down the current plotting device

Description:

Close down the current plotting device, storing the current picture description in the AGI database. The position and world coordinate limits of the plot will be stored in the AGI database, along with any comment text.

ENDPLOT must be executed before using a plotting application from another package (*e.g.* KAPPA) – failure to do so will result in an AGI error and may corrupt the AGI database.

This command is an ICL hidden procedure which uses the undocumented PONGO application `ENDPONGO`.

Usage:

```
endplot [comment]
```

Parameters:

COMMENT = _CHAR (Read and Write)

A comment for the AGI database entry for the plot that has just been completed. Any comment will be prefixed with the string "PONGO:". ["Final viewport"]

ERASE
Clear the graphics screen

Description:

The plotting surface is cleared.

This command is a synonym for `CLEAR SCREEN=YES`.

Usage:

```
erase
```

ERRORBAR
Draw error bars on the plotted data

Description:

Draw error bars in the X or Y directions, either treating the values in the EXCOL and EYCOL data areas as symmetric errors about the point, or as an upper limit with the XCOL or YCOL data area holding the other limit.

PONGO will plot error bars correctly even after logarithms of the data have been taken for the symmetric option, as long as the `CLOG` application has been used to perform the transformation (as opposed to `CCMATH`). For the non-symmetric case, the `CCMATH` application should be used to take the logarithms of the data in the EXCOL and EYCOL data areas.

Usage:

```
errorbar action [erterm]
```

Parameters:

ACTION = _CHAR (Read)

"X" or "Y" depending upon which set of error bars is to be drawn.

[The value is prompted for.]

ERTERM = _REAL (Read and Write)

The length of the terminals on the error bars: a multiple of the default length.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

SYMERR = _LOGICAL (Read and Write)

If TRUE, the values in the error data areas represent a symmetric error about the values in the data columns. If FALSE, the data columns represent the lower limits, and the error columns represent the upper limits. [TRUE]

ERRX
Draw symmetric error bars in the X direction

Description:

Draw symmetric error bars in the X direction.

This command is a synonym for `ERRORBAR X`.

Usage:

```
errx [erterm]
```

Parameters:

ERTERM = _REAL (Read and Write)

The length of the terminals on the error bars: a multiple of the default length.

If no value is specified on the command line, the current value is used. The current value is initially set to 1.0.

ERRY
Draw symmetric error bars in the Y direction

Description:

Draw symmetric error bars in the Y direction.

This command is a synonym for `ERRORBAR Y`.

Usage:

```
erry [erterm]
```

Parameters:

ERTERM = _REAL (Read and Write)

The length of the terminals on the error bars: a multiple of the default length.

If no value is specified on the command line, the current value is used. The current value is initially set to 1.0.

EXCOLUMN
Specify the column containing the X-axis error data

Description:

Specify the column in the data file from which the errors on the X-axis data are to be read.

This command is a synonym for SETGLOBAL PONGO_EXCOL.

Usage:

```
excolumn excol
```

Parameters:

EXCOL = _CHAR (Write)

The column number (counting from 1), or the symbolic name of a column, from which the X-axis error data are read by the `READF` command. The value "0" means "do not read these data". []

EXPAND
Set the character height

Description:

Change the character height scaling. This command scales the default character height and also alters the size of the tick marks and symbols that PONGO plots. The default character height is about 1/40 of the viewport height.

This command is a synonym for `CHANGE CHEIGHT=`.

Usage:

```
expand cheight
```

Parameters:

CHEIGHT = _REAL (Read and Write)

The character height scaling. This parameter scales the default character height and also alters the size of the tick marks and symbols that PGPLOT plots. The default character height in PGPLOT is about 1/40 of the viewport height. []

EYCOLUMN
Specify the column containing the Y-axis error data

Description:

Specify the column in the data file from which the errors on the Y-axis data are to be read.

This command is a synonym for SETGLOBAL PONGO_EYCOL.

Usage:

```
eycolumn eycol
```

Parameters:

EYCOL = _CHAR (Write)

The column number (counting from 1), or the symbolic name of a column, from which the Y-axis error data are read by the `READF` command. The value `"0"` means `"`do not read these data`"`. []

FILLSTY
Change fill-style plotting attributes

Description:

Change the PGPLOT fill-style attributes.

This is an hidden ICL procedure that makes use of the CHANGE command.

Usage:

```
fillsty fillsty [angle] [separation] [phase]
```

Parameters:

ANGLE = _REAL (Read)

The angle hatch lines make with horizontal, in degrees. Only used when FILLSTY is 3. [45.0]

FILLSTY = _INTEGER (Read)

The fill style used by PGPLOT. The fill styles are as follows:

- 1 – solid fill,
- 2 – hollow fill,
- 3 – hatched,
- 4 – cross-hatched

The must be specified on the command line.

PHASE = _REAL (Read)

The fraction of SEPARATION that hatched lines are displaced. Modifying this between regions makes their separation more obvious. This is only used when FILLSTY is 3.

SEPARATION = _REAL (Read)

The separation of hatched lines. The unit spacing is 1 percent of the smaller of the height or width of the view surface. Only used when FILLSTY is 3.

FITCURVE
Fit a curve to the data

Description:

This command fits data using either a polynomial or a smooth spline. If weights are available these can be used when determining the fit.

The resultant fit parameters are displayed and written to the environment for use by the PLOTFUN application, which can re-plot the polynomial or spline with other datasets *etc.*

Usage:

```
fitcurve action npoly [weight]
```

Parameters:

ACTION = _CHAR (Read and Write)

Type of curve to be fitted. Currently this action must be "POLY" or "SPLINE"

If the parameter is not specified on the command line, it will be prompted for.

NPOLY = _INTEGER (Read and Write)

The order of the polynomial or spline. For polynomials this can range from 1 to 9 and for splines from 2 to 6.

If the parameter is not specified on the command line, it will be prompted for.

COLOUR = _INTEGER (Read and Write)

The colour index used when plotting the fitted curve.

If no value is specified on the command line, the current value is used. If there is no current value, a default of 2 (*i.e.* red) will be used.

WEIGHT = _LOGICAL (Read and Write)

Whether the fit is to use the y-axis error data in the EYCOL data area, if available. If no error data are available, the fit will always be unweighted.

If the value is not specified on the command line, the current value is used. If there is no current value, a default value of TRUE is used.

POLYCOEF[10] = _DOUBLE (Write)

If ACTION is "POLY", the polynomial coefficients resulting from the fit are written to this parameter.

The value of this parameter is written to the global parameter PONGO_POLYCOEF.

POLYFILE = _CHAR (Read)

The name of a file to contain the polynomials coefficients of the fit (only used ACTION is "POLY"). This file can be used to store the fit permanently and can be used by the PLOTFUN command to redraw the fit.

[POLYFILE.dat]

SMOOTH = _REAL (Read)

Only used if ACTION is "SPLINE". This factor determines the tradeoff between the closeness and smoothness of the spline fit. It should be a real number greater than 0.0. Normally this is dynamically defaulted to a number equal to the number of points to be fitted. Decreasing this value to 0.0 produces an interpolating spline fit.

SPLINEFILE = _CHAR (Read)

The name of a file to contain the coefficients and knot positions from the spline fit – used when ACTION is "SPLINE". This file can be used by the PLOTFUN command to redraw the fit.

[The value of the global parameter PONGO_SPLINEF is used. If PONGO_SPLINEF is not defined, the value is prompted for.]

XMIN = _REAL (Read)

The minimum X value to be used in the fit.

The value of the global parameter PONGO_XMIN is used. If PONGO_XMIN is not defined, the default value 0.0 is used.

XMAX = _REAL (Read)

The maximum X value to be used in the fit.

The value of the global parameter PONGO_XMAX is used. If PONGO_XMAX is not defined, the default value 1.0 is used.

Notes:

- This routine fits a general polynomial of the form:
  y = $a1 + a2 * x + a3 * x^2 + a4 * x^3 .....an * x^n$
  which has order n-1, using a least squares approach.

- The spline fit is characterised by the positions of the knots and the spline coefficients (and the order used for the splines) all of which are stored in the spline file. The number of knots used in the fit can only be influenced by using the SMOOTH parameter. Higher values of this give larger smoothing factors. A value of 0 gives an interpolating spline fit.

FITLINE
Fit a straight line to the data

Description:

If the y-axis error data are available and the WEIGHT parameter is TRUE, this application will perform a weighted least squares fit to a straight line for the data over the range delimited by the XMIN and XMAX parameters. If the y-axis error data are not available or WEIGHT is FALSE, an unweighted fit is performed. The best fit straight line is plotted. The resultant fit parameters are displayed along with some simple statistics for the data (these statistics are also weighted in the case of a weighted fit).

Usage:

```
fitline [colour]
```

Parameters:

COLOUR = _INTEGER (Read and Write)

The colour index used when plotting the fitted line.

If the value is not specified on the command line, the current value is used. If there is no current value, a default value of 2 (*i.e.* red) is used.

WEIGHT =_LOGICAL (Read and Write)

Whether the fit is to use the y-axis error data in the EYCOL data area, if available. If no error data are available, the fit will always be unweighted.

If the value is not specified on the command line, the current value is used. If there is no current value, a default value of TRUE is used.

XMIN = _REAL (Read)

The minimum X value to be used in the fit.

The value of the global parameter PONGO_XMIN is used. If PONGO_XMIN is not defined, the default value 0.0 is used.

XMAX = _REAL (Read)

The maximum X value to be used in the fit.

The value of the global parameter PONGO_XMAX is used. If PONGO_XMAX is not defined, the default value 1.0 is used.

FONT
Set the text font

Description:

Change the text font.

This command is a synonym for `CHANGE FONT=`.

Usage:

```
font font
```

Parameters:

FONT = _INTEGER (Read and Write)

The font used by PGPLOT. The styles for each font are as follows:

- 1 – single-stroke font,
- 2 – roman font,
- 3 – italic font,
- 4 – script font.

[]

GETPOINT
Retrieve information for a plotted data point

Description:

Return the attributes of a plotted data point to ICL variables.

This application has been written to aid the implementation of ICL procedures. Because it is only possible to make enquiries about a single point per invocation, any ICL procedure built around GETPOINT will work slowly if a large number of data are involved. For such cases it may be better to consider writing a customised PONGO application.

Usage:

```
getpoint action value
```

Parameters:

ACTION = _CHAR (Read)

The method of specifying the data point in question. If "N", interpret the VALUE parameter as specifying the index number of that point. If "C", the VALUE parameter is used to try to match the LABCOL entry for a point.

[The value is prompted for.]

VALUE = _CHAR (Read)

The value to be used in the search for the data point. Depending upon the value of the ACTION parameter, this may either be an integer specifying the index number of the point in the data area, or a case-sensitive minimum match string for a label column entry in the data area.

[The value is prompted for.]

X = _REAL (Write)

The returned value of the X coordinate of the selected point.

Y = _REAL (Write)

The returned value of the Y coordinate of the selected point.

Z = _REAL (Write)

The returned value of the Z coordinate of the selected point.

EX = _REAL (Write)

The returned value of the X coordinate error of the selected point.

EY = _REAL (Write)

The returned value of the Y coordinate error of the selected point.

SYMBOL = _INTEGER (Write)

The returned value of the symbol column of the selected point.

LABEL = _CHAR (Write)

The returned value of the label column of the selected point.

Examples:

```
PONGO> GETPOINT C '3C45' X=(XP) Y=(YP)
```

This will return the X and Y coordinates of the data point that has the label '3C45', if it exists, to the ICL variables XP and YP.

GPOINTS
Plot points or lines between the data

Description:

General plotting application. This application can be used simply to plot a symbol at the position of each point, to plot a symbol whose size depends upon the values in the ZCOL data area, or to draw lines connecting the data points.

Usage:

```
gpoints action [symbol]
```

Parameters:

ACTION = _CHAR (Read)

The type of plot to produce. This can be "C", "P" or "S", with the following meanings:

- "C" (connect) – This action simply draws straight line segments between the data points.
- "P" (points) – Draw a symbol at each of the data points. The symbol type that is used to mark each point is determined in one of 3 ways:

  - If no SYMBOL parameter is supplied on the command line, and no symbol numbers have been read into the symbol data area by `READF`, the point style will be set by the current value of SYMBOL.
  - If no SYMBOL parameter is supplied on the command line, and values have been read into the symbol data area by `READF`, the symbol number for each point will determine the style of the point plotted.
  - If SYMBOL is specified on the command line, it will override each of the above options. The same specified symbol will be used to mark all points.

  The value of the symbol index should be an integer which refers to the standard PGPLOT symbols.
- "S" (sizeplot) – This action uses the values stored in the ZCOL data area to determine the size of the plotted symbol. The value of each entry in the ZCOL data area is effectively used as an argument to a `CHANGE CHEIGHT` command before each point is plotted. The SCALE parameter can be used to make these values cover a reasonable range by multiplying the Z data values.

[The value is prompted for.]

SYMBOL = _INTEGER (Read and Write)

The PGPLOT symbol number that is used to mark the data points.

If a value is specified on the command line, it will be used for plotting symbols for all the data. If not value is specified on the command line, the application attempts to use the SYMCOL data for its symbols. If no symbol values have been read into the SYMCOL data area, the current value is used for all the data. The current value is initially set to 1.

SCALE = _REAL (Read and Write)

The scale factor used to multiply the ZCOL data values to get a reasonable range of symbol sizes when ACTION="S".

If no value is specified on the command line, the current value is used. The current value is initially set to 1.0.

PROJECTION = _CHAR (Read)

Specifies the geometry that is to be used to plot the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

GRID
Draw a coordinate grid at specified intervals

Description:

Draw a grid in the current projection at specified intervals in spherical coordinates. The intervals, start and end values should all be specified in degrees. The defaults for the grid separations normally produce desirable effects for all sky plots. In specifying the grid intervals it is sometimes necessary to take account of rounding errors that might occur, and to bear in mind that in some geometries a single point on the celestial sphere maps onto two points on the projected coordinates – some care is needed to ensure that the whole grid is drawn.

Usage:

```
grid [phimin] [phimax] [phistep] [themin] [themax] [thestep]
```

Parameters:

PHIMIN = _DOUBLE (Read and Write)

The start longitude in degrees for the coordinate grid.

If no value is specified on the command line, the current value is used. The current value is initially set to 0.0.

PHIMAX = _DOUBLE (Read and Write)

The end longitude in degrees for the coordinate grid.

If no value is specified on the command line, the current value is used. The current value is initially set to 360.0.

PHISTEP = _DOUBLE (Read and Write)

The spacing between longitude grid lines in degrees.

If no value is specified on the command line, the current value is used. The current value is initially set to 30.0.

THEMIN = _DOUBLE (Read and Write)

The start latitude in degrees for the coordinate grid.

If no value is specified on the command line, the current value is used. The current value is initially set to -90.0.

THEMAX = _DOUBLE (Read and Write)

The end latitude in degrees for the coordinate grid.

If no value is specified on the command line, the current value is used. The current value is initially set to 90.0.

THESTEP = _DOUBLE (Read and Write)

The spacing between latitude grid lines in degrees.

If no value is specified on the command line, the current value is used. The current value is initially set to 10.0.

PROJECTION = _LITERAL (Read)

The geometry to be used to plot the grid. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _LITERAL (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _LITERAL (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

Notes:

- It is sometimes necessary to specify the grid intervals in a manner which avoids rounding errors to obtain the last grid line. e.g 9.9999 instead of 10. You may also need to sometimes draw the grid twice using say 30.00001 as well as 29.99999 to get this final line.

GT_CIRCLE
Draw a great circle between two points

Description:

Draw a great circle between two points in current projection. The great circle is specified by giving the coordinates in degrees of two points on the celestial sphere. Either the small or large great circle may be drawn.

Usage:

```
gt_circle [phistart] [thestart] [phiend] [theend]
```

Parameters:

PHISTART = _DOUBLE (Read and Write)

The longitude of the start of the great circle in degrees.

If no value is specified on the command line, the current value is used. The current value is initially set to 0.0.

THESTART = _DOUBLE (Read and Write)

The latitude of the start of the great circle in degrees.

If no value is specified on the command line, the current value is used. The current value is initially set to 0.0

PHIEND = _DOUBLE (Read and Write)

The longitude of the end of the great circle in degrees.

If no value is specified on the command line, the current value is used. The current value is initially set to 0.0.

THEEND = _DOUBLE (Read and Write)

The latitude of the end of the great circle in degrees.

If no value is specified on the command line, the current value is used. The current value is initially set to 0.0.

ACUTE = _LOGICAL (Read and Write)

If TRUE, the smaller great circle arc is drawn between the given points.

If no value is specified on the command line, the current value is used. The current value is initially set to TRUE.

PROJECTION = _CHAR (Read)

The geometry to be used to plot the great circle. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

HISTOGRAM
Bin and plot a histogram of the data

Description:

The data in the `XCOL` data area are binned and plotted as a histogram. It is possible to plot several histograms with different bin sizes from the same data in `XCOL` because the data are unaffected by HISTOGRAM.

This command is a synonym for `PLOTHIST H`.

Usage:

```
histogram [binmin] [binmax] [nbin]
```

Parameters:

BINMIN = _REAL (Read and Write)

The lower limit of the binning.

If no value is specified on the command line, the current value is used. If there is no current value, the value of the global parameter PONGO_XMIN is used.

BINMAX = _REAL (Read and Write)

The upper limit of the binning.

If no value is specified on the command line, the current value is used. If there is no current value, the value of the global parameter PONGO_XMAX.

NBIN = _INTEGER (Read and Write)

The number of equally sized bins to be drawn between the limits of the histogram.

If no value is specified on the command line, the current value is used. The current value is initially set to 10.

AUTOSCALE = _LOGICAL (Read and Write)

If TRUE, PGPLOT auto-scaling is used to determine the plotting limits. If FALSE, the limits defined by the bins of the histogram determine the plotting limits. Here, the plotting limits must previously have been set using the LIMITS application and the plot frame drawn using `BOXFRAME`. Setting NOAUTOSCALE can be used to draw more than one histogram on the same plot.

If no value is specified on the command line, the current value is used. The current value is initially set to TRUE.

FILL = _LOGICAL (Read)

When ACTION is `"H"` this parameter controls whether the histogram will be drawn filled with the current fill, colour and hatching styles. [FALSE]

INQUIRE
Display PONGO status information

Description:

Display information about the status of PONGO and the data which have been read in. The options are:

- PGPLOT – Display the current font, character height colour *etc.*
- LIMITS – Display the data limits and the PGPLOT world coordinate limits.
- COLUMNS – Display the column names from the data file, if they have been set up appropriately.
- DEVICES – Display the available graphics devices.
- DATA – Show the data that has been read in.

More than one of these options can be specified on the command line at any one time.

The DATA option uses additional parameters to allow scrolling.

Usage:

```
inquire
```

Parameters:

PGPLOT = _LOGICAL (Read)

Display the current PGPLOT plotting attributes. [FALSE]

LIMITS = _LOGICAL (Read)

Display the data limits and the current PGPLOT viewport and world coordinate limits. [FALSE]

COLUMNS = _LOGICAL (Read)

Display the data file column headings (if available). [FALSE]

DEVICES = _LOGICAL (Read)

Display the plotting devices available. [FALSE]

DATA = _LOGICAL (Read)

Display the contents of all data areas in a formatted form. [FALSE]

PAGE = _INTEGER (Read and Write)

The page length of the terminal (in the range 1 to 100). It is used to stop information scrolling off the top of the screen. The parameter will be prompted for at the end of each screen: a null response to the prompt will terminate the listing.

If no value is specified on the command line, the current value is used. The current value is initially set to 24.

FROM = _INTEGER (Read and Write)

The number of the first item to be displayed.

If no value is specified on the command line, the current value is used. The current value is initially set to 0 (implying the start of the list).

TO = _INTEGER (Read and Write)

The number of the last item to be displayed.

If no value is specified on the command line, the current value is used. The current value is initially set to 0 (implying the end of the list).

LABCOLUMN
Specify the column used for data labels

Description:

Specify the column in the data file from which the data labels are to be read.

This command is a synonym for SETGLOBAL PONGO_LABCOL.

Usage:

```
labcolumn labcol
```

Parameters:

LABCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the symbolic name for each data point is read by the `READF` command. The value `"0"` means `"`do not read these data`"`. []

LABEL
Draw the axis labels and title on the plot

Description:

Draw the axis labels and the title on the plot. If the COLUMNS parameter is specified, the axis labels are obtained from the column labels read from the data file.

Usage:

```
label [xlabel] [ylabel] [title]
```

Parameters:

XLABEL = _CHAR (Read and Write)

The X-axis label.

If no value is specified on the command line, then if COLUMNS is TRUE the value is taken from the column heading in the data file, otherwise the value of the global parameter PONGO_XLABEL is used. If PONGO_XLABEL is not defined, the value `" "` is used.

YLABEL = _CHAR (Read and Write)

The Y-axis label.

If no value is specified on the command line, then if COLUMNS is TRUE the value is taken from the column heading in the data file, otherwise the value of the global parameter PONGO_YLABEL is used. If PONGO_YLABEL is not defined, the value `" "` is used.

TITLE = _CHAR (Read and Write)

The plot title.

[The value of the global parameter PONGO_TITLE is used. If PONGO_TITLE is not defined, the value `" "` is used.]

COLUMNS = _LOGICAL (Read)

If TRUE, the values of the X and Y labels will be obtained from the column headings in the data file. [FALSE]

LIMITS
Set the world coordinate limits

Description:

The world coordinate limits are set from the parameters given on the command line.

This command is a synonym for `WORLD GIVEN`.

Usage:

```
limits [xmin] [xmax] [ymin] [ymax]
```

Parameters:

XMIN = _REAL (Read and Write)

The world coordinate of the left-hand edge of the plot.

[The value of the global parameter PONGO_XMIN is used.]

XMAX = _REAL (Read and Write)

The world coordinate of the right-hand edge of the plot.

[The value of the global parameter PONGO_XMAX is used.]

YMIN = _REAL (Read and Write)

The world coordinate of the lower edge of the plot.

[The value of the global parameter PONGO_YMIN is used.]

YMAX = _REAL (Read and Write)

The world coordinate of the upper edge of the plot.

[The value of the global parameter PONGO_YMAX is used.]

PROJECTION = _CHAR (Read and Write)

The geometry to be used for plotting the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

[The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.]

RACENTRE = _CHAR (Read and Write)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.]

DECCENTRE = _CHAR (Read and Write)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.]

LTYPE
Set the line style

Description:

Change the line style.

This command is a synonym for `CHANGE LINESTY=`.

Usage:

```
ltype linesty
```

Parameters:

LINESTY = _INTEGER (Read and Write)

The line style used by PGPLOT. The line style may be one of the following:

- 1 – full line,
- 2 – dashed,
- 3 – dot-dash-dot-dash,
- 4 – dotted,
- 5 – dash-dot-dot-dot.

[]

LWEIGHT
Set the line width

Description:

Change the line width scaling (the normal line width is 1.0).

This command is a synonym for `CHANGE LINEWID=`.

Usage:

```
lweight linewid
```

ADAM Parameters :

LINEWID = _INTEGER (Read and Write) The line width scaling. This parameter scales the default line width. []

MARK
Draw a point mark at the specified position

Description:

Draw a point mark at a specified position.

This command is a synonym for `PRIM K`.

Usage:

```
mark x y [symbol]
```

Parameters:

X = _REAL (Read and Write)

The X coordinate of the point.

[The value is prompted for.]

Y = _REAL (Read and Write)

The Y coordinate of the point.

[The value is prompted for.]

SYMBOL = _INTEGER (Read and Write)

The PGPLOT symbol number for drawing the point mark.

If no value is given on the command line, the current value is used. The current value is initially set to 1.

PROJECTION = _CHAR (Read)

The geometry that is to be used for plotting. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

Notes:

When using non-planar coordinates, the coordinates should be given in degrees.

MOVE
Set the current pen position

Description:

The current pen position is set to the given coordinates.

This command is a synonym for `PRIM M`.

Usage:

```
move x y
```

Parameters:

X = _REAL (Read and Write)

The X coordinate of the point.

If no value is specified on the command line, the value is prompted for.

Y = _REAL (Read and Write)

The Y coordinate of the point.

If no value is specified on the command line, the value is prompted for.

PROJECTION = _CHAR (Read)

The geometry that is to be used for plotting. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".This parameter is only relevant when marking points.

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

MTEXT
Draw a text string relative to the viewport

Description:

Draw a text string on the plot at a position specified relative to the viewport. The command uses the PGPLOT routine PGMTEXT.

This command is a an ICL hidden procedure which uses the `WTEXT` application.

Usage:

```
mtext side xpos ypos justification text
```

Parameters:

SIDE = _CHAR (Read and Write)

The side of the viewport where the text is to plotted. This may be one of the following:

- "T" – The top edge.
- "B" – The bottom edge.
- "L" – The left-hand edge.
- "R" – The right-hand edge.
- "LV" – The left-hand edge, but with the string written vertically.
- "RV" – The right-hand edge, but with the string written vertically.

[]

XPOS = _REAL (Read and Write)

The number of character heights from the viewport where the text is to be plotted (negative values are allowed). []

YPOS = _REAL (Read and Write)

The fraction along the edge where the text is to be plotted. []

JUSTIFICATION = _REAL (Read and Write)

The justification about the specified point (in the range 0.0 to 1.0). Here, 0.0 means left justify the text relative to the data point, 1.0 means right justify the text relative to the data point, 0.5 means centre the string on the data point, other values will give intermediate justifications. []

TEXT = _CHAR (Read and Write)

The text string to be plotted. This may include any of the PGPLOT control sequences for producing special characters. []

PALETTE
Change the plotting pen colours

Description:

The colour representation for a given colour index is updated.

Usage:

```
palette [colour] [red] [green] [blue]
```

Parameters:

COLOUR = _INTEGER (Read and Write)

The colour index, *i.e.* pen, to be updated (in the range 0 to 255).

If no value is specified on the command line, the current value is used. The current value is initially set to 1.

RED = _REAL (Read and Write)

The red intensity (in the range 0.0 to 1.0).

If no value is specified on the command line, the current value is used. The current value is initially set to 1.0.

GREEN = _REAL (Read and Write)

The green intensity (in the range 0.0 to 1.0).

If no value is specified on the command line, the current value is used. The current value is initially set to 1.0.

BLUE = _REAL (Read and Write)

The blue intensity (in the range 0.0 to 1.0).

If no value is specified on the command line, the current value is used. The current value is initially set to 1.0.

PAPER
Change the size and aspect ratio of the plotting surface

Description:

The width of the plotting surface and its aspect ratio (*i.e.* height/width) are modified.

Usage:

```
paper width aspect
```

Parameters:

WIDTH = _REAL (Read and Write)

The width of the plotting surface in inches. If the specified width is 0.0, the maximum possible width for the device is used.

[The value is prompted for.]

ASPECT = _REAL (Read and Write)

The aspect ratio of the plotting surface: *i.e.* height/width.

[The value is prompted for.]

PCOLUMN
Specify the column used for symbol codes

Description:

Specify the column in the data file from which the symbol codes are to be read.

This command is a synonym for SETGLOBAL PONGO_SYMCOL.

Usage:

```
pcolumn symcol
```

Parameters:

SYMCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the PGPLOT symbol code for each data point is read by the `READF` command. The value `"0"` means `"`do not read these data`"`. []

PEN
Set the current pen

Description:

Set the current pen used for plotting.

This command is a synonym for `CHANGE COLOUR=`.

Usage:

```
pen colour
```

Parameters:

COLOUR = _INTEGER (Read and Write)

The pen number (colour index) PGPLOT uses for plotting. The value should be between 0 and 255. Usually the first 16 pens are predefined to have the following colours:

- 0 – background,
- 1 – foreground (default),
- 2 – red,
- 3 – green,
- 4 – blue,
- 5 – cyan,
- 6 – magenta,
- 7 – yellow,
- 8 – red + yellow (orange),
- 9 – green + yellow,
- 10 – green + cyan,
- 11 – blue + cyan,
- 12 – blue + magenta,
- 13 – red + magenta,
- 14 – dark grey,
- 15 – light grey.

It is possible to change the colour representation of any of the pen colour indices using the PALETTE application. []

PLOTFUN
Plot a given function

Description:

Plot a function specified on the command line by a Fortran-like expression (using TRANSFORM, SUN/61), or through the parameters resulting from a previous fit (using a polynomial or spline).

Usage:

```
plotfun action expression [xmin] [xmax]
```

Parameters:

ACTION = _CHAR (Read)

The type of function to be plotted. This must be one of the following:

- "FUNC" – Use a Fortran-like expression to define the function.
- "POLY" – Use a set of polynomial coefficients to define the function.
- "SPLINE" – Use a set of spline coefficients from the file SPLINEFILE to define the function.

[The value is prompted for.]

EXPRESSION = _CHAR (Read)

The Fortran-like expression to be plotted, in terms of X.

[The value is prompted for.]

XMIN = _REAL (Read)

The value of X from which the function is plotted.

[The value of the global parameter PONGO_XMIN is used.]

XMAX = _REAL (Read)

The value of X to which the function is plotted.

[The value of the global parameter PONGO_XMAX is used.]

INFILE = _LOGICAL (Read)

Used when ACTION is "POLY". If TRUE then the polynomial coefficients are stored in a file (the first line of which is the order of the polynomial), otherwise the coefficients are given using the POLYCOEF parameter.

[FALSE]

NPOLY = _INTEGER (Read and Write)

The order of the polynomial: used when ACTION is "POLY".

[The value of the global parameter PONGO_NPOLY is used.]

POLYCOEF = _DOUBLE (Read)

A list of polynomial coefficients: used when ACTION is "POLY".

[The value of the global parameter PONGO_POLYCOEF is used.]

POLYFILE = _CHAR (Read)

The name of a file containing the polynomial coefficients to be plotted. Only used when ACTION is "POLY" and INFILE is TRUE.

[POLYFILE.dat]

SPLINEFILE = FILENAME (Read)

The name of the file containing the coefficients and knot positions from a previous spline fit – used when ACTION is "SPLINE".

[The value of the global parameter PONGO_SPLINEF is used. If PONGO_SPLINEF is not defined, the value is prompted for.]

PLOTHIST
Plot a histogram of the data

Description:

This application has two modes:

- Bin the data in the XCOL data area and plot the result.
- Plot data that have already been binned and provided in the XCOL and YCOL data areas.

Usage:

```
plothist action [binmin] [binmax] [nbin]
```

Parameters:

ACTION = _CHAR (Read)

The mode of PLOTHIST as described above:

- "H" – If the data in the XCOL data area are not binned, they can be binned and then plotted. It is possible to plot several histograms with different bin sizes from the same data in XCOL because the data are unaffected by PLOTHIST.
- "B" – If the data have already been binned, this mode will plot a histogram using the XCOL and YCOL data areas. The YCOL data area should specify the bin edges and the YCOL data area should contain their respective frequencies.

[The value is prompted for.]

BINMIN = _REAL (Read and Write)

When ACTION is "H", this parameter specifies the lower limit of the binning.

If no value is specified on the command line, the current value is used. If there is no current value, the value of the global parameter PONGO_XMIN is used.

BINMAX = _REAL (Read and Write)

When ACTION is "H", this parameter specifies the upper limit of the binning.

If no value is specified on the command line, the current value is used. If there is no current value, the value of the global parameter PONGO_XMAX.

FILL = _LOGICAL (Read)

When ACTION is "H" this parameter controls whether the histogram will be drawn filled with the current fill, colour and hatching styles. [FALSE]

NBIN = _INTEGER (Read and Write)

When ACTION is "H", this parameter specifies the number of equally sized bins to be drawn between the limits of the histogram.

If no value is specified on the command line, the current value is used. The current value is initially set to 10.

AUTOSCALE = _LOGICAL (Read and Write)

When ACTION is "H" this parameter specifies whether PGPLOT auto-scaling is used to determine the plotting limits. If FALSE, the limits defined by the bins of the histogram determine the plotting limits. Here, the plotting limits must previously have been set using the LIMITS application and the plot frame drawn using `BOXFRAME`. Setting NOAUTOSCALE can be used to draw more than one histogram on the same plot.

If no value is specified on the command line, the current value is used. The current value is initially set to TRUE.

CENTRE = _LOGICAL (Read)

When ACTION is "B", this parameter specifies whether the values in the XCOL data area denote the centre of each bin (when TRUE) or its lower edge (when FALSE). [FALSE]

POINTS
Draw a point mark at each of the data points

Description:

Draw a symbol at each of the data points. The symbol type that is used to mark each point is determined in one of three ways:

- If no SYMBOL parameter is supplied on the command line, and no symbol numbers have been read into the symbol data area by `READF`, the point style will be set by the current value of SYMBOL.
- If no SYMBOL parameter is supplied on the command line, and values have been read into the symbol data area by `READF`, the symbol number for each point will determine the style of the point plotted.
- If SYMBOL is specified on the command line, it will override each of the above options. The same specified symbol will be used to mark all points.

The value of the symbol index should be an integer which refers to the standard PGPLOT symbols.

This command is a synonym for `GPOINTS P`.

Usage:

```
points [symbol]
```

Parameters:

SYMBOL = _INTEGER (Read and Write)

The PGPLOT symbol number that is used to mark the data points.

If a value is specified on the command line, it will be used for plotting symbols for all the data. If no value is specified on the command line, the application attempts to use the SYMCOL data for its symbols. If no symbol values have been read into the SYMCOL data area, the current value is used for all the data. The current value is initially set to 1.

PROJECTION = _CHAR (Read)

Specifies the geometry that is to be used to plot the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

PRIM
Perform primitive plotting functions

Description:

Perform the primitive plotting functions: move to, draw line to, and mark.

Usage:

```
prim action x y [symbol]
```

Parameters:

ACTION = _CHAR (Read and Write)

Type of primitive function. This may be one of the following:

- "M" – move to,
- "D" – draw line to,
- "K" – mark.

[The value is prompted for.]

X = _REAL (Read and Write)

X coordinate of point.

[The value is prompted for.]

Y = _REAL (Read and Write)

Y coordinate of point.

[The value is prompted for.]

SYMBOL = _INTEGER (Read and Write)

PGPLOT symbol number for drawing the point mark.

If no value is specified on the command line, the current value is used. The current value is initially set to 1.

PROJECTION = _CHAR (Read)

The geometry that is to be used for plotting. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".This parameter is only relevant when marking points.

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

Notes:

When using non-planar coordinates, the coordinates should be given in degrees. Lines drawn between points are straight. Use GT_CIRCLE for drawing lines that follow great circles.

PTEXT
Draw a text string at the specified position and angle

Description:

Draw a text string on the plot at the specified position and angle. The command uses the PGPLOT routine PGPTEXT.

This command is a an ICL hidden procedure which uses the `WTEXT` application.

Usage:

```
ptext xpos ypos angle justification text
```

Parameters:

XPOS = _REAL (Read and Write)
The X coordinate of the text. []

YPOS = _REAL (Read and Write)
The Y coordinate of the text. []

ANGLE = _REAL (Read and Write)
The angle relative to the horizontal at which the text string is to be plotted. []

JUSTIFICATION = _REAL (Read and Write)
The justification about the specified point (in the range 0.0 to 1.0). Here, 0.0 means left justify the text relative to the data point, 1.0 means right justify the text relative to the data point, 0.5 means centre the string on the data point, other values will give intermediate justifications. []

TEXT = _CHAR (Read and Write)
The text string to be plotted. This may include any of the PGPLOT control sequences for producing special characters. []

PTINFO
Get the coordinates of a specified data point

Description:

The value of a label in the LABCOL data area is given and the resulting (x,y) coordinates associated with the label are printed.

This command is an ICL hidden procedure which uses the GETPOINT application.

Usage:

```
ptinfo label
```

Parameters:

LABEL = _CHAR (Read)

The label associated with the data point. []

PVECT
Draw proper motion vectors

Description:

Draw proper motion vectors on a projection of the celestial sphere. The XCOL and YCOL data areas are assumed to contain positions in radians, the EXCOL and EYCOL data areas are assumed to contain the proper motions in radians per year. It is possible to use the ERSCALE parameter to multiply the the proper motion so that it is correct for a given number of years. (The proper motion in RA is assumed to be $\dot{\alpha}\cos\delta$.)

Usage:

```
pvect [erscale]
```

Parameters:

ERSCALE = _REAL (Read and Write)

The scale factor for multiplying the vectors.

[The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.]

ZMULT = _LOGICAL (Read)

If TRUE, the ZCOL values are additionally used to multiply the vectors. [FALSE]

PROJECTION = _CHAR (Read)

The geometry used to plot the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

RADIATE
Draw a line from the given point to the first NP data points

Description:

A line is drawn from the given radiant, position (X,Y), to the first NP data points in the XCOL and YCOL data areas.

This command is an ICL hidden procedure which uses successive calls to the `GETPOINT`, `MOVE` and `DRAW` applications.

Usage:

```
radiate x y np
```

Parameters:

X = _REAL (Read)
The X-axis position of the radiant. []

Y = _REAL (Read)
The Y-axis position of the radiant. []

NP = _INTEGER (Read)
The number of data in the XCOL and YCOL data areas to use, starting from the beginning of the data. []

READF
Read from a formatted data file

Description:

Read input data from a formatted data file. READF attempts to read data from columns in an sequential formatted file in the most flexible manner possible. It is possible to specify the following:

- What comment delimiter characters are used within the data file – if used, the comment delimiter must be the first character of a line in the data file.
- What the column delimiters are (more than one character is possible).
- Symbolic names for each of the data columns.

The application is intended to be very robust: if a read error occurs within a line, READF will report an error and attempt to continue.

The application has many parameters for controlling how data are read, but the default values of these parameters are sufficient for reading most data files.

Usage:

```
readf [data] [all=?]
        { selcond=?
    all
```

Parameters:

DATA = FILENAME (Read and Write)

The name of the formatted data file.

If the value is not specified on the command line, the value of the global parameter PONGO_DATA is used. If PONGO_DATA is not defined, the current value is used. If the current value is not defined, the value is prompted for.

HARDCOM = _CHAR (Read and Write)

A character used to indicate a comment line in the data file. The character must appear in the first column of a comment.

If the value is not specified on the command line, the current value is used. The current value is initially set to "!".

SOFTCOM = _CHAR (Read and Write)

A character to indicate a comment line in the data file. This parameter allows a second character to be used as a comment delimiter. The character must appear in the first column of a comment.

If the value is not specified on the command line, the current value is used. The current value is initially set to "!".

XCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the X-axis data are read. The value "0" means "do not read these data".

If the value is not specified on the command line, the value of the global parameter PONGO_XCOL is used. If PONGO_XCOL is not defined, the current value is used. The current value is initially set to "0".

YCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the Y-axis data are read. The value "0" means "do not read these data".

If the value is not specified on the command line, the value of the global parameter PONGO_YCOL is used. If PONGO_YCOL is not defined, the current value is used. The current value is initially set to "0".

ZCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the Z-axis data are read. The value "0" means "do not read these data".

If the value is not specified on the command line, the value of the global parameter PONGO_ZCOL is used. If PONGO_ZCOL is not defined, the current value is used. The current value is initially set to "0".

EXCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the X-axis error data are read. The value "0" means "do not read these data".

If the value is not specified on the command line, the value of the global parameter PONGO_EXCOL is used. If PONGO_EXCOL is not defined, the current value is used. The current value is initially set to "0".

EYCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the Y-axis error data are read. The value "0" means "do not read these data".

If the value is not specified on the command line, the value of the global parameter PONGO_EYCOL is used. If PONGO_EYCOL is not defined, the current value is used. The current value is initially set to "0".

LABCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the symbolic name for each data point is read. The value "0" means "do not read these data".

If the value is not specified on the command line, the value of the global parameter PONGO_LABCOL is used. If PONGO_LABCOL is not defined, the current value is used. The current value is initially set to "0".

SYMCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the PGPLOT symbol code for each data point is read. The value "0" means "do not read these data".

If the value is not specified on the command line, the value of the global parameter PONGO_SYMCOL is used. If PONGO_SYMCOL is not defined, the current value is used. The current value is initially set to "0".

DELIM = _CHAR (Read and Write)

The character string interpreted as a column delimiter when reading the data file. For example, this can be used to read LATEX format tables by setting DELIM="&".

If the value is not specified on the command line, the current value is used. The current value is initially set to " ".

FROM = _INTEGER (Read and Write)

The first line of data to be read from the data file. The value 0 defaults to the beginning of the file. [0]

TO = _INTEGER (Read and Write)

The last line of data to be read from the data file. The value 0 defaults to the end of the file. [0]

SELCOND = _CHAR (Read and Write)

A condition (or criterion) upon which to select values from the data file. This condition has the form

[SELECT_COL] [COND] [VAL1{,VAL2, ...}]

where

- SELECT_COL is the data area used for the selection test. This can be specified either by column number (counting from 1) or by the symbolic name of a column. There is no restriction on which column is used for selection, *i.e.* it does not have to be one of the columns from which data are being read.
- COND is the selection criterion. It may be one of the following:
  - "=" – equals;
  - "#" – not equal;
  - ">" – greater than;
  - "<" – less than;
  - "CE" – equal to string;
  - "C#" – not equal to string;
  - "RA" – in the range VAL1 to VAL2;
  - "LI" – select if in the following list of values;
  - "EX" – exclude if in the following list of values;
  - "IN" – select if the substring is contained within the value;
  - "A>" – absolute value greater than;
  - "A<" – absolute value less than.
- VAL1{,VAL2, ...} the value (or values) against which the selection is made.

Note that there must be white space around the selection criterion. A value of "0" means "read everything".

[The value is prompted for.]

XOPT = _CHAR (Read and Write)
A string that controls the style of the X-axis labelling and tick marks. It consists of a series of letters, which are described fully in the documentation for the BOXFRAME application.

READF updates the value of the global parameters PONGO_XOPT. The application will automatically remove any "L" characters at the start of the options string, because it is assumed that they have been inserted by the CLOG application – any new data will not have had logarithms taken. If data are given in logarithmic form, the "L" character should be inserted into the options strings anywhere except at the start.

[The value of the global parameter PONGO_XOPT is used. If PONGO_XOPT is not defined, the default value "BCNST" is used.]

YOPT = _CHAR (Read and Write)
A string that controls the style of the Y-axis labelling and tick marks. It consists of a series of letters, which are described fully in the documentation for the BOXFRAME application.

READF updates the value of the global parameters PONGO_YOPT. The application will automatically remove any "L" characters at the start of the options string, because it is assumed that they have been inserted by the CLOG application – any new data will not have had logarithms taken. If data are given in logarithmic form, the "L" character should be inserted into the options strings anywhere except at the start.

[The value of the global parameter PONGO_YOPT is used. If PONGO_XOPT is not defined, the default value "BCNST" is used.]

ERSCALE = _REAL (Read)
The scale factor to be applied to the EXCOL and EYCOL data.

[The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.]

ADD = _LOGICAL (Read)
If FALSE, the data values already held will be cleared before reading new data; if TRUE, the data read will be appended to the existing data. [FALSE]

ALL = _LOGICAL (Read and Write)
If TRUE, the whole data file will be read; if FALSE, a selection condition will be prompted for.

If the value is not specified on the command line, the current value is used. The current value is initially set to TRUE.

QUICK = _LOGICAL (Read and Write)
If TRUE, a "quick mode" read is performed. This mode can only be used on files which exclusively contain numeric data. This parameter can over-ride the action of the LABCOL and SELCOND parameters.

If the value is not specified on the command line, the current value is used. The current value is initially set to FALSE.

NDATA = _INTEGER (Write)
The number of data read from the data file. [0]

RESETPONGO
Reset the state of PONGO.

Description:

The state of PONGO (global parameter values, AGI and PGPLOT) is returned to a default state: all global parameters are reset to their default values, the AGI database for the graphics device is cleared, and the viewport is reset to the standard PGPLOT viewport. The global parameter values for the PONGO data file (PONGO_DATA) and the polynomial coefficients (PONGO_POLYCOEF) are not reset.

This command is an ICL hidden procedure using the SETGLOBAL command and the `CHANGE`, `CLEAR`, `VIEWPORT` and `WORLD` applications.

Usage:

```
resetpongo
```

RTODEG
Convert the specified data area from radians to degrees

Description:

The values in the specified data area are converted from radians to degrees.

This command is an ICL hidden procedure which uses the `CCMATH` application.

Usage:

```
rtodeg column
```

Parameters:

COLUMN = _CHAR (Read and Write)

The column to be converted from radians to degrees. []

SETPROJ
Set the projection geometry

Description:

Set the projection related global variables. These are then used by all the relevant applications.

This is an hidden ICL procedure that makes use of the SETGLOBAL command.

Usage:

```
setproj projection [racentre] [deccentre]
```

Parameters:

PROJECTION = _CHAR (Write)

The geometry that is to be used. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter must be specified on the command line. It sets the value of the global parameter PONGO_PROJECTN.

RACENTRE = _CHAR (Write)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter sets the value of the PONGO_RACENTRE global variable.

DECCENTRE = _CHAR (Write)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter sets the value of the PONGO_DECCENTRE global variable.

SHOWPONGO
Show the PONGO global parameters

Description:

The PONGO global parameters are listed with their current values using the ICL command GETGLOBAL. It does not list the polynomial coefficients, PONGO_POLYCOEF.

This command is an ICL hidden procedure using the GETGLOBAL command.

Usage:

```
showpongo
```

SIZEPLOT
Draw point marks of differing sizes at each of the data points

Description:

Draw symbols of differing sizes at each of the data points. This application uses the values stored in the ZCOL data area to determine the size of each plotted symbol; *i.e.* the value of each entry in the ZCOL data area is effectively used as an argument to a `CHANGE CHEIGHT` command before each point is plotted. The SCALE parameter can be used to make these values cover a reasonable range by multiplying the Z data values.

This command is a synonym for `GPOINTS S`.

Usage:

```
sizeplot [symbol]
```

Parameters:

SYMBOL = _INTEGER (Read and Write)

The PGPLOT symbol number that is used to mark the data points.

If a value is specified on the command line, it will be used for plotting symbols for all the data. If no value is specified on the command line, the application attempts to use the SYMCOL data for its symbols. If no symbol values have been read into the SYMCOL data area, the current value is used for all the data. The current value is initially set to 1.

SCALE = _REAL (Read and Write)

The scale factor used to multiply the ZCOL data values to get a reasonable range of symbol sizes.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

PROJECTION = _CHAR (Read)

Specifies the geometry that is to be used to plot the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

This parameter is not specified on the command line. The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.

RACENTRE = _CHAR (Read)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.

DECCENTRE = _CHAR (Read)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

This parameter is not specified on the command line. The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.

SYMCOLUMN
Specify the column used for symbol codes

Description:

Specify the column in the data file from which the symbol codes are to be read.

This command is a synonym for SETGLOBAL PONGO_SYMCOL.

Usage:

```
symcolumn symcol
```

Parameters:

SYMCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the PGPLOT symbol code for each data point is read by the `READF` command. The value `"0"` means `"`do not read these data`"`. []

TEXT
Draw a text string on the plot at the specified position

Description:

Draw a text string on the plot at coordinates (x,y), using the PGPLOT routine PGTEXT.

This command is a synonym for `WTEXT S`.

Usage:

```
text xpos ypos text
```

Parameters:

XPOS = _REAL (Read and Write)

The X coordinate of the text.

[The value is prompted for.]

YPOS = _REAL (Read and Write)

The Y coordinate of the text.

[The value is prompted for.]

TEXT = _CHAR (Read and Write)

The text string to be plotted. This may include any of the PGPLOT control sequences for producing special characters.

[The value is prompted for.]

VECT
Draw vectors from each data point

Description:

Use the values in the EXCOL and EYCOL data areas as signed offsets in X and Y to plot vectors from the data points. These vectors are scaled using the error scale parameter ERSCALE. Individual vectors may also be scaled using the contents of the ZCOL data area.

Usage:

```
vect [erscale] [zmult]
```

Parameters:

ERSCALE = _REAL (Read and Write)

Factor for scaling all vectors.

[The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.]

ZMULT = _LOGICAL (Read)

If TRUE, the values of the ZCOL data area are used as multipliers for the vectors. This parameter may only be specified on the command line. [FALSE]

VIEWPORT
Set the viewport for the current plotting device

Description:

Control the PGPLOT viewport. The viewport is the region of the plotting surface through which the graph is seen.

Usage:

```
viewport action
    { [xvpmin] [xvpmax] [yvpmin] [yvpmax]
  action
```

Parameters:

ACTION = _CHAR (Read)

The method used to set the viewport. It may be one of the following:

- "STANDARD" – The viewport is set to the standard PGPLOT viewport.
- "ADJUST" – The viewport is adjusted so that the scales along the X and Y axes are the same number of world coordinate units per unit length. The newly created viewport fits within the old viewport.
- "NDC" – The viewport is set by specifying its extent in the X and Y directions in terms of normalised device coordinates (*i.e.* coordinates that run from 0 to 1 along the horizontal and vertical directions).
- "INCHES" – The viewport is set by specifying its extent in the X and Y directions in terms of inches.

[The value is prompted for.]

XVPMIN = _REAL (Read and Write)

The left hand side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

XVPMAX = _REAL (Read and Write)

The right hand side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

YVPMIN = _REAL (Read and Write)

The lower side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

YVPMAX = _REAL (Read and Write)

The upper side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

XMIN = _REAL (Read)

The left hand edge of the world coordinate system.

[The value of the global parameter PONGO_XMIN is used. If PONGO_XMIN is not defined, the default value 0.0 is used.]

XMAX = _REAL (Read)

The right hand edge of the world coordinate system.

[The value of the global parameter PONGO_XMAX is used. If PONGO_XMAX is not defined, the default value 1.0 is used.]

YMIN = _REAL (Read)

The lower edge of the world coordinate system.

[The value of the global parameter PONGO_YMIN is used. If PONGO_YMIN is not defined, the default value 0.0 is used.]

YMAX = _REAL (Read)

The upper edge of the world coordinate system.

[The value of the global parameter PONGO_YMAX is used. If PONGO_YMIN is not defined, the default value 1.0 is used.]

VPORT
Set the viewport using normalised device coordinates

Description:

The viewport is set by specifying its extent in the X and Y directions in terms of normalised device coordinates (*i.e.* coordinates that run from 0 to 1 along the horizontal and vertical directions).

This command is a synonym for `VIEWPORT NDC`.

Usage:

```
vport [xvpmin] [xvpmax] [yvpmin] [yvpmax]
```

Parameters:

XVPMIN = _REAL (Read and Write)

The left hand side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

XVPMAX = _REAL (Read and Write)

The right hand side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

YVPMIN = _REAL (Read and Write)

The lower side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

YVPMAX = _REAL (Read and Write)

The upper side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

VP_BH
Set the viewport to the bottom half of the plotting surface

Description:

Set the viewport to the bottom half of the plotting surface. If requested space around the viewport will be reserved for any labels.

This command is a procedure based on "VIEWPORT NDC".

Usage:

```
vp_bh [labels]
```

VP_BL
Set the viewport to the bottom-left quarter of the plotting surface

Description:

Set the viewport to the bottom left quarter of the plotting surface. If requested space around the viewport will be reserved for any labels.

This command is a procedure based on "VIEWPORT NDC".

Usage:

```
vp_bl [labels]
```

VP_BR
Set the viewport to the bottom-right quarter of the plotting surface

Description:

Set the viewport to the bottom right quarter of the plotting surface. If requested space around the viewport will be reserved for any labels.

This command is a procedure based on "VIEWPORT NDC".

Usage:

```
vp_br [labels]
```

VP_TH
Set the viewport to the top half of the plotting surface

Description:

Set the viewport to the top half of the plotting surface. If requested space around the viewport will be reserved for any labels.

This command is a procedure based on "VIEWPORT NDC".

Usage:

```
vp_th [labels]
```

VP_TL
Set the viewport to the top-left quarter of the plotting surface

Description:

Set the viewport to the top left quarter of the plotting surface. If requested space around the viewport will be reserved for any labels.

This command is a procedure based on "VIEWPORT NDC".

Usage:

```
vp_tl [labels]
```

VP_TR
Set the viewport to the top-right quarter of the plotting surface

Description:

Set the viewport to the top right quarter of the plotting surface. If requested space around the viewport will be reserved for any labels.

This command is a procedure based on "VIEWPORT NDC".

Usage:

```
vp_tr [labels]
```

VSIZE
Set the viewport using its physical size in inches

Description:

The viewport is set by specifying its minimum and maximum extents in the X and Y directions in terms of inches.

This command is a synonym for `VIEWPORT INCHES`.

Usage:

```
vsize [xvpmin] [xvpmax] [yvpmin] [yvpmax]
```

Parameters:

XVPMIN = _REAL (Read and Write)

The left hand side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

XVPMAX = _REAL (Read and Write)

The right hand side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

YVPMIN = _REAL (Read and Write)

The lower side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

YVPMAX = _REAL (Read and Write)

The upper side of the viewport.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

VSTAND
Set the standard viewport

Description:

The viewport is set to the standard PGPLOT viewport.

This command is a synonym for `VIEWPORT STANDARD`.

Usage:

```
vstand
```

WNAD
Adjust the viewport so that the X and Y scales are the same

Description:

The viewport is adjusted so that the scales along the X and Y axes are the same number of world coordinate units per unit length. The newly created viewport fits within the old viewport.

This command is a synonym for `VIEWPORT ADJUST`.

Usage:

```
wnad
```

Parameters:

XMIN = _REAL (Read)

The left hand edge of the world coordinate system.

The value of the global parameter PONGO_XMIN is used. If PONGO_XMIN is not defined, the default value 0.0 is used.

XMAX = _REAL (Read)

The right hand edge of the world coordinate system.

The value of the global parameter PONGO_XMAX is used. If PONGO_XMAX is not defined, the default value 1.0 is used.

YMIN = _REAL (Read)

The lower edge of the world coordinate system.

The value of the global parameter PONGO_YMIN is used. If PONGO_YMIN is not defined, the default value 0.0 is used.

YMAX = _REAL (Read)

The upper edge of the world coordinate system.

The value of the global parameter PONGO_YMAX is used. If PONGO_YMIN is not defined, the default value 1.0 is used.

WORLD
Set the world coordinates for the plot

Description:

Set up the world coordinate limits for the plot. The world coordinate system is the one in which the data are plotted. It is possible to specify the limits explicitly, or to have them calculated from the range of the data that have been read, or to recall the limits from a previous plot using AGI (SUN/48).

Usage:

```
world action
      { [xmin] [xmax] [ymin] [ymax]
      {
      { piclab=?
   action
```

Parameters:

ACTION = _CHAR (Read and Write)

The way in which the world coordinate limits are to be determined:

- "DATA" – The limits are calculated from the data limits, with a small border added.
- "DATA0" – The limits are calculated as for DATA, but the origin is included as one of the data points.
- "GIVEN" – The limits specified on the command line are used.
- "RECALL" – The coordinates are recalled from a previous plot using the AGI database.

[The value is prompted for.]

XMIN = _REAL (Read and Write)

The world coordinate of the left-hand edge of the plot.

The application will determine the value if ACTION is one of "DATA", "DATA0" or "RECALL". If ACTION="GIVEN" and no value is specified on the command line, the value of the global parameter PONGO_XMIN is used.

XMAX = _REAL (Read and Write)

The world coordinate of the right-hand edge of the plot.

The application will determine the value if ACTION is one of "DATA", "DATA0" or "RECALL". If ACTION="GIVEN" and no value is specified on the command line, the value of the global parameter PONGO_XMAX is used.

YMIN = _REAL (Read and Write)

The world coordinate of the lower edge of the plot.

The application will determine the value if ACTION is one of "DATA", "DATA0" or "RECALL". If ACTION="GIVEN" and no value is specified on the command line, the value of the global parameter PONGO_YMIN is used.

YMAX = _REAL (Read and Write)

The world coordinate of the upper edge of the plot.

The application will determine the value if ACTION is one of "DATA", "DATA0" or "RECALL". If ACTION="GIVEN" and no value is specified on the command line, the value of the global parameter PONGO_YMAX is used.

PICLAB = _CHAR (Read and Write)

The AGI label of the picture to be recalled.

[If ACTION="RECALL", the value is prompted for.]

PROJECTION = _CHAR (Read and Write)

The geometry to be used for plotting the data. This is explained in more detail in the section on projections. Allowed values: "NONE", "TAN", "SIN", "ARC", "GLS", "AITOFF", "MERCATOR" and "STG".

[The value of the global parameter PONGO_PROJECTN is used. If PONGO_PROJECTN is not defined, the default value "NONE" is used.]

RACENTRE = _CHAR (Read and Write)

The centre of the projection in RA (*i.e.* the angle must be specified as hh:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_RACENTRE is used. If PONGO_RACENTRE is not defined, the default value "0" is used.]

DECCENTRE = _CHAR (Read and Write)

The centre of the projection in declination (*i.e.* the angle must be specified as dd:mm:ss.sss). This parameter is only required for PROJECTION values other than "NONE".

[The value of the global parameter PONGO_DECCENTRE is used. If PONGO_DECCENTRE is not defined, the default value "0" is used.]

ERSCALE = _REAL (Read)

The scale factor to be applied to the EXCOL and EYCOL data when determining the limits of the world co-ordinates. ERSCALE is used only when ACTION = "DATA" or "DATA0". [The value of the global parameter PONGO_ERSCALE is used. If PONGO_ERSCALE is not defined, the default value 1.0 is used.]

WRITEI
Write information to an output file

Description:

Write specified information concerning the current data-set to an output file.

Usage:

```
writei action file
```

Parameters:

ACTION = _CHAR (Read and Write)

The type of information to be written. This may be one of the following:

- "LABLST" – Write the internal list of labels out.
- "DATA" – Write out selected data.
- "AGIPIC" – Write the label, name and comment for the current AGI picture to the AGI database.

[The value is prompted for.]

FILE = FILENAME (Read and Write)

The name of the output file to be written.

[The value is prompted for.]

FORMAT = _CHAR (Read and Write)

The Fortran FORMAT to be used.

If the value is not specified on the command line, the current value is used. The current value is initially set to "G25.16".

AGINAME = _CHAR (Read and Write)

The AGI name for the current picture. This may be one of the following:

- "DATA" – Used to indicate that the AGI picture contains the representation of data in some graphical form (*i.e.* a graph).
- "FRAME" – Used to indicate that the AGI picture contains a group of other plots (*i.e.* several "DATA" pictures).

["DATA"]

AGICOMMENT = _CHAR (Read and Write)

The AGI comment for the current picture.

If the value is not specified on the command line, the current value is used. The current value is set to "User viewport".

AGILABEL = _CHAR (Read and Write)

The AGI label for the current picture.

[The value is prompted for.]

X = _LOGICAL (Read and Write)

If TRUE, the XCOL data area will be output. [FALSE]

Y = _LOGICAL (Read and Write)

If TRUE, the YCOL data area will be output. [FALSE]

Z = _LOGICAL (Read and Write)

If TRUE, the ZCOL data area will be output. [FALSE]

EX = _LOGICAL (Read and Write)

If TRUE, the EXCOL data area will be output. [FALSE]

EY = _LOGICAL (Read and Write)

If TRUE, the EYCOL data area will be output. [FALSE]

WTEXT
Draw a text string on the plot

Description:

Draw a text string on the current plot at a given position, justification and orientation.

Usage:

```
wtext action xpos ypos text [side] [justification] [angle]
```

Parameters:

ACTION = _CHAR (Read and Write)

The way in which the text string is to be written. It may be one of the following:

- "P" – Use PGPTEXT which allows the position, justification and angle of the text to be specified.
- "M" – Use PGMTEXT which allows the text to be written relative to the viewport.
- "S" – Use PGTEXT which allows only simple (x,y) positioning of the text.

[The value is prompted for.]

XPOS = _REAL (Read and Write)

If ACTION is "P" or "S", the X coordinate of the text. With the "M" action, this parameter specifies the number of character heights from the viewport where the text is to be plotted (negative values are allowed).

[The value is prompted for.]

YPOS = _REAL (Read and Write)

If ACTION is "P" or "S", the Y coordinate of the text. With the "M" action, this parameter specifies the fraction along the edge where the text is to be plotted.

[The value is prompted for.]

TEXT = _CHAR (Read and Write)

The text string to be plotted. This may include any of the PGPLOT control sequences for producing special characters.

[The value is prompted for.]

SIDE = _CHAR (Read and Write)

If ACTION="M", the side of the viewport where the text is to plotted. This may be one of the following:

- "T" – The top edge.
- "B" – The bottom edge.
- "L" – The left-hand edge.
- "R" – The right-hand edge.
- "LV" – The left-hand edge, but with the string written vertically.
- "RV" – The right-hand edge, but with the string written vertically.

If the value is not specified on the command line, the current value is used. The current value is initially set to "T".

JUSTIFICATION = _REAL (Read and Write)

The justification about the specified point (in the range 0.0 to 1.0). Here, 0.0 means left justify the text relative to the data point, 1.0 means right justify the text relative to the data point, 0.5 means centre the string on the data point, other values will give intermediate justifications.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

ANGLE = _REAL (Read and Write)

If ACTION="P", the angle relative to the horizontal at which the text string is to be plotted.

If the value is not specified on the command line, the current value is used. The current value is initially set to 0.0.

XCOLUMN
Specify the column containing the X-axis data

Description:

Specify the column in the data file from which the X-axis data are to be read.

This command is a synonym for SETGLOBAL PONGO_XCOL.

Usage:

```
xcolumn xcol
```

Parameters:

XCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the X-axis data are read by the `READF` command. The value `"0"` means "do not read these data`"`. []

XERR
Draw symmetric error bars in the X direction

Description:

Draw symmetric error bars in the X direction.

This command is a synonym for `ERRORBAR X`.

Usage:

```
xerr [erterm]
```

Parameters:

ERTERM = _REAL (Read and Write)

The length of the terminals on the error bars: a multiple of the default length.

If the value is not specified on the command line, the current value is used. The current value is initially set to 1.0.

XLINEAR
Put 1 ... N into the XCOL data area

Description:

Put the row number (*i.e.* 1 to NDATA) into the XCOL data area.

This command is a synonym for `CCMATH X=INDEX`.

Usage:

```
xlinear
```

XLOGARITHM
Take the logarithm of the X-axis data

Description:

Take the base 10 logarithm of the X-axis data.

This command is a synonym for `CLOG X`.

Usage:

```
xlogarithm
```

XOFFSET
Add a constant offset to the X-axis data

Description:

Add a constant offset to the X-axis data.

This command is an ICL hidden procedure which uses the `CCMATH` application.

Usage:

```
xoffset offset
```

Parameters:

OFFSET = _REAL (Read)

The value of the X-axis offset. []

XSCALE
Multiply the values in the XCOL and EXCOL data areas by a constant

Description:

Multiply the values in the XCOL and EXCOL data areas by a constant.

This command is an ICL hidden procedure which uses the `CCMATH` application.

Usage:

```
xscale scale
```

Parameters:

SCALE = _REAL (Read)

The constant by which the XCOL and EXCOL data areas are scaled. []

YCOLUMN
Specify the column containing the Y-axis data

Description:

Specify the column in the data file from which the Y-axis data are to be read.

This command is a synonym for SETGLOBAL PONGO_YCOL.

Usage:

```
ycolumn ycol
```

Parameters:

YCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the Y-axis data are read. The value "0" means "do not read these data". []

YERR
Draw symmetric error bars in the Y direction

Description:

Draw symmetric error bars in the Y direction.

This command is a synonym for `ERRORBAR Y`.

Usage:

```
yerr [erterm]
```

Parameters:

ERTERM = _REAL (Read and Write)

The length of the terminals on the error bars: a multiple of the default length.

If no value is specified on the command line, the current value is used. The current value is initially set to 1.0.

YLINEAR
Put 1 ... N into the YCOL data area

Description:

Put the row number (*i.e.* 1 to NDATA) into the YCOL data area.

This command is a synonym for `CCMATH Y=INDEX`.

Usage:

```
ylinear
```

YLOGARITHM
Take the logarithm of the Y-axis data

Description:

Take the base 10 logarithm of the Y-axis data.

This command is a synonym for `CLOG Y`.

Usage:

```
ylogarithm
```

YOFFSET
Add a constant offset to the Y-axis data

Description:

Add a constant offset to the Y-axis data.

This command is an ICL hidden procedure which uses the `CCMATH` application.

Usage:

```
yoffset offset
```

Parameters:

OFFSET = _REAL (Read)

The value of the Y-axis offset. []

YSCALE
Multiply the values in the YCOL and EYCOL data areas by a constant

Description:

Multiply the values in the YCOL and EYCOL data areas by a constant.

This command is an ICL hidden procedure which uses the CCMATH application.

Usage:

```
yscale scale
```

Parameters:

SCALE = _REAL (Read)

The constant by which the YCOL and EYCOL data areas are scaled. []

ZCOLUMN
Specify the column containing the Z-axis data

Description:

Specify the column in the data file from which the Z-axis data are to be read.

This command is a synonym for SETGLOBAL PONGO_ZCOL.

Usage:

```
zcolumn zcol
```

Parameters:

ZCOL = _CHAR (Read and Write)

The column number (counting from 1), or the symbolic name of a column, from which the Z-axis data are read. The value "0" means "do not read these data". []

ZSCALE
Multiply the values in the ZCOL data area by a constant

Description:

Multiply the values in the ZCOL data area by a constant.

This command is an ICL hidden procedure which uses the CCMATH application.

Usage:

```
zscale scale
```

Parameters:

SCALE = _REAL (Read)

The constant by which the ZCOL data area is scaled. []

C PONGO and the MONGO Graphics Package

PONGO has been designed to have broadly the same command interface as that offered by the MONGO[6] graphics package. A number of aliases (ICL DEFSTRING definitions) have been set up for certain commands so that PONGO will do more or less what is expected for the equivalent MONGO command, but there are some substantial differences. It might well be possible to do a better job of imitating MONGO using ICL procedures; however, the aim is not to imitate the precise behaviour of MONGO, but for existing users of MONGO to be met with a package that is not totally unfamiliar to them when they begin using PONGO. Ultimately, PONGO offers substantially more than MONGO, both by its flexibility and by what it can do.

Having read the rest of this document it will have become clear that there are a substantial number of differences between MONGO and PONGO: perhaps most significantly, it is not possible to run a MONGO script file and expect it to work. The most important difference at the command level is that once XCOL *etc.* have been set up, the data must be input explicitly using the READF command, where MONGO would read the data in each time that a command that needs them is executed. *Table 2* is a list of the commands that work as expected (with the above proviso) and *Table 3* gives a list of the closest equivalents to MONGO commands.

D Notes for IRAF/CL users

The most noticable difference in the way that PONGO behaves and "normal" IRAF tasks is that it uses its own display and graphic devices. To display on an X device the usual way of referring to a window is by the name xw. This is as used thoughout this document. If you need to change the size of this window then use the command (from the shell after closing the existing window):

```
% xmake xwindows -width xxx -height xxx
```

You should not change the size of an X window after it is created. The X window is created using the GWM services which you can read more about in SUN/130 (which, for instance, describes other command-line options and how to set up your X resources to define the window size). If you want to plot to a postscript file then there are a range of possible devices, the most usual of which are pscol_p (colour postscript portrait mode) and epsfcol_p (colour encapsulated postscript). All the devices available can be viewed using the INQUIRE DEVICES=YES command.

The way that PONGO behaves when run from CL tends to be slightly different to that indicated in the full routine descriptions. The main changes are that some parameters will not behave as described and there are very few of what are described as "global parameters". The on-line help available in CL should show what the actual parameters associated with a command are (you can, of course, also view these using the lparam command).

The commands described as "aliases" and "synonyms" previously are actually CL scripts most of which will work pretty much as described. One problem to beware of is that several such commands do not have any associated parameters (*i.e.* ADVANCE, CONNECT, ERASE, RESETPONGO,

[6] The MONGO package was once available at all Starlink nodes, at the time of writing this is no longer the case, so this reference, indeed this whole section, is now largely historical.

Command	Behavioural differences
BOX	The optional arguments are different in the two cases.
CONNECT	
DATA	
DRAW	
ERASE	
ERRORBAR	The way in which the errors are plotted is different.
LTYPE	The actual styles produced are different.
EXPAND	In PONGO this will also alter sizes of tick-marks.
LWEIGHT	
PCOLUMN	The symbol numbers in the file refer to the standard PGPLOT marker symbols.
PEN	
POINTS	PONGO has additional optional arguments.
XCOLUMN	In PONGO a symbolic name can be used optionally for the column description.
XLINEAR	In PONGO no arguments can be given. The array is always filled in an increasing integer sequence from one. It is possible to perform any desired manipulation on the values in the column using the `CCMATH` command.
XLOGARITHM	
YCOLUMN	In PONGO a symbolic name can be used optionally for the column description.
YLINEAR	In PONGO no arguments can be given. The array is always filled in an increasing integer sequence from one (see the description of `XLINEAR`).
YLOGARITHM	

Table 2: Commands which have nearly the same effect in MONGO and PONGO

MONGO	PONGO	Behaviour
AXIS	BOXFRAME	Most effects achievable by can be accomplished by altering the parameters of the BOXFRAME command.
BADY	READF	Use the COND parameter to selectively read the file.
ECOLUMN	EXCOLUMN	The errors for the X and Y directions are read separately in PONGO.
	EYCOLUMN	The errors for the X and Y directions are read separately in PONGO.
GRID	BOXFRAME	A grid can be drawn by using G in XOPT (not to be confused with PONGO command GRID).
HARDCOPY		
HISTOGRAM	PLOTHIST	Use the command PLOTHIST B (note that the command PLOTHIST H will automatically bin unbinned data).
LABEL	WTEXT	PONGO is more flexible.
LINES	READF	Use the FROM and TO parameters.
PTYPE	POINTS	PONGO is restricted to the standard PGPLOT symbols: although there are 32 of them, you cannot define your own as you can in MONGO.
PUTLABEL	WTEXT	

Table 3: PONGO command equivalents for certain MONGO commands

SHOWPONGO, VSTAND, WNAD, XLINEAR, XLOGARITHM, YLINEAR and YLOGARITHM) so cannot be setup as part of a mkscript session.

One unusual characteristic of PONGO commands is that some have what are described as "Write" or "Unused" parameters. Both these types cannot be set using eparam (or more precisely any values you assign to these will be ignored). The "Write" parameters will be modified when the command completes and are a way of getting values back from the command for other uses (the NDATA parameter of READF is one example, this is set to the actual number of points read from the file).

If you want to know more about how Starlink packages operate from IRAF/CL then see the document SUN/217.

E Changes

E.1 This release (2.0-2)

PONGO now uses native PGPLOT for plotting rather than the deprecated Starlink GKS PGPLOT. This should fix plotting and interoperability problems with recent versions of KAPPA.

E.2 Previous release (2.0-1)

PONGO has been considerably extended to now work from the IRAF/CL command language. To use this load the pongo package. The commands that are available are the same as when running from ICL.

Other changes of note are:

- The READF command has been changed so that '@' is no longer needed when using an ICL variable to store the file name. This is INCOMPATIBLE with previous behaviour and any scripts using the '@' form will need changing, i.e. lines with statements like:

  ```
  READF DATA=('@'&FILE)
  ```

  Should be changed to:

  ```
  READF DATA=(FILE)
  ```

- An example has been added to the panic section to show how PONGO can be used to plot more than 5000 points at a time (5000 is the current maximum number of points that can be read at the same time).
- READF has been fixed so that it is possible to once again read data from more than one file.
- Errors in the documentation that still included descriptions of the previous Chebyshev version of FITLINE have been corrected.
- A description of how to close the edges of grids has been added to the GRID command section. This command now also attempts to complete latitude lines to their end points (this reduces the ragged appearance at the ends of latitude lines).

- The YLINEAR command now works!
- The VP_xx commands have been modified to accept an argument which indicates that space around the viewport should be reserved for any labels.
- FITCURVE now uses the XMIN and XMAX parameters as described (polynomial curves previously used all the data). The YMAX and YMIN parameters are no longer used. It also now writes any polynomial coefficients to a file (this can be read by PLOTFUN).
- FITCURVE has also had a memory control bug fixed.
- CURSE now also reports approximate sky coordinates when reading the cursor positions.
- A new command FILLSTY has been introduced. This conveniently sets the fill-style attributes (for polygons and histograms).
- A new command DRAWPOLY has been introduced. This draws polygons that can be filled.
- A new command SETPROJ has been introduced. This sets the projection geometry related global variables (under CL this sets all the parameters of the projection aware commands).
- The PRIM command now allows projections for all its actions (this also effects the commands, MOVE and DRAW). Note that the lines drawn are straight-lines on the plot surface, not great circles. The GT_CIRCLE command already supplies this ability.
- The commands XSCALE, YSCALE, YOFFSET and XOFFSET now allow negative factors.

E.3 Release 1.3

PONGO has been updated to fix several problems with the BEGPLOT command. This now correctly selects the last AGI DATA picture when OVERLAY is set (so that for instance KAPPA images can be labelled), and avoids problems when selecting pictures using labels (which resulted in a runaway device open error condition).

If ICL is exited before closing PONGO (using the ENDPLOT command) a warning is now issued and an attempt to repair the AGI database is now made.

A bug in the ANNOTATE command has been fixed. This command now correctly handles the RACENTRE and DECCENTRE parameters.

All PONGO command descriptions now include a usage section. This shows the order of any positional parameters explicitly.

A new example procedure has been added to show the use of the VECT and PVECT commands and how to label astrometric projections. The AGI example has been expanded to show the results.

The routines PLOTFUN and FITCURVE have been changed to remove the NAG dependency of PONGO.

The FITCURVE command now fits a general, rather than a Chebyshev polynomial and now has the ability to fit B-splines. Both these functions can be re-plotted by the PLOTFUN command.

E.4 Release 1.2

The ability to plot labels of the form `HH MM SS.S` and `DD MM SS.S` has been added to the `BOXFRAME` command.

To use this just read in your positions in radians (the PONGO command `READF` already does this for you if the positions are in `HH:MM:SS`, `DD:MM:SS` format, or the `DEGTOR` command will convert columns from degrees to radians) and then use a command like:

```
ICL> BOXFRAME XOPT='BCNSTZHG' YOPT='BCNSTZDG'
```

This now uses the PGPLOT routine PGTBOX. `BOXFRAME` has also been modified to allow the use of the "1" and "2" option flags which force the labels to be all decimal or exponential format.

Other changes are:

(1) The `CHANGE` command now supports hatched fill styles and sets the text background colour.

(2) The `INQUIRE` command reports the hatched fill style and text background colour.

(3) `PLOTHIST` histograms can now be filled.