PGPLOT — Graphics Subroutine Library
v5.1.1/v5.2.0
User's Guide

Abstract

PGPLOT is the *de facto* standard plotting library in Astronomy. Starlink supports both the original 'Native' version and the GKS-based 'Starlink' version. This document describes their use on Starlink systems.

Contents

1   Introduction

PGPLOT is a high level graphics package for plotting X *versus* Y plots, functions, histograms, bar charts, contour maps and images. Complete diagrams can be produced with a minimal number of subroutine calls, but control over colour, lines-style, character font, *etc,*. is available if required. The package was written (by Dr T J Pearson of the Caltech astronomy department) with astronomical applications in mind and has become a *de facto* standard for graphics in astronomy world wide.

The package exists in two versions: the original or 'Native' version which uses the low level graphics package GRPCKG which was also written at Caltech, and a version developed by Starlink, in collaboration with Dr Pearson, which uses GKS. The two versions have identical subroutine interfaces and in most cases, applications can be moved from one version to the other simply by re-linking.

Starlink supports both the 'Native' version and the 'Starlink' version which co-exist on Starlink systems. Most packages available in the Starlink Software Collection currently use the Starlink version but work is currently underway to replace this with the Native version.

The current Native PGPLOT is version 5.2.0. The current Starlink PGPLOT is based on PGPLOT Version 5.1.1.

The main source of information on using PGPLOT is the PGPLOT manual. The manual for version 5 is still in preparation but a hypertext version can be found at:

        http://astro.caltech.edu/~tjp/pgplot/

This describes the original GRPCKG *Native-PGPLOT* version but the majority of the manual applies equally to both versions.

When using the GKS *Starlink-PGPLOT* version the following sections of the manual should be ignored and the information in this user note used instead:

- Using PGPLOT (chapter 1)

- Graphics Devices (chapter1)

- Compiling and Running the Program (chapter 2)

- Appendix D — Supported Devices

- Appendix E — Writing a Device Handler

- Appendix F — Installation Instructions

- Appendix G — Porting PGPLOT

2   Native-PGPLOT

Readers interested in the Starlink-PGPLOT version should refer to section 3.

2.1   Using the Native version

To link a program with the Native-PGPLOT library, use the following:

```
% f77 prog.f -L/star/lib `pgplot_link`
```

It is currently not possible to use Native PGPLOT with ADAM applications though in future this facility will be provided.

2.2   Native-PGPLOT device names

Native-PGPLOT device names are described in the PGPLOT manual. The Native-PGPLOT device names are not the same as those used for other Starlink graphics and it does not support the Starlink device name system. However, substantially the same graphics device types are available (Xwindows, Postscript).

It should be noted that the two PGPLOT systems have different Xwindows device drivers which do not interact – hence a program using Starlink-PGPLOT and drawing to the 'xwindows' device will not draw in the same display window as a Native-PGPLOT program drawing to the '/XWIN' device.

2.3   Native-PGPLOT examples

> *On non-Starlink systems you may have to replace /star by some other path name to locate the files referred to in this section.*

Binaries of the example programs can be found in `/star/bin/examples/pgplot`. They can be run with a command such as:

```
/star/bin/examples/pgplot/pgdemon
```

where n is between 1 and 17, provided that they have been installed. (They may not have been in order to save disk space).

3   Starlink-PGPLOT

Readers interested in the Native-PGPLOT version should refer to section 2.

3.1   Using the GKS version

There are two ways in which the GKS version of PGPLOT can be used:

(1) As a self contained graphics package where *all* graphics, including opening and closing the workstation, is done with PGPLOT. Programs written in this way can be run with other implementations of PGPLOT. Such programs are linked with the command:

```
% f77 prog.f -L/star/lib `pgp_link`
```

(2) To plot a picture in the current viewport of an already open GKS workstation (see section 3.4).

To use PGPLOT in an ADAM application, consult the *ADAM Graphics Programmer's Guide* (SUN/113).

3.2   Starlink-PGPLOT device names

Any graphics device supported by GKS can be used with Starlink PGPLOT and device names are translated using the graphics name service described in *GNS – Graphics Name Service* (SUN/57). Device names containing a '/' character (such as UNIX file names) must be surrounded by quote ('"') characters.

If a question mark is typed in response to the prompt from PGBEG, a list of those workstation names defined on your system will be listed on the terminal.

On some hard copy devices the output from a PGPLOT program is a file and some further action (such as printing the file) is required to produce a plot. If you are unfamiliar with a particular device, consult SUN/83.

If one of the metafile workstations is selected the metafile can be tailored for a particular real workstation type by appending /TARGET=*workstation* to the device specification. The default target is the monochrome A4 Postscript workstation. The resulting metafile can be played back on any workstation but will be tailored with respect to such things as resolution, number of colour indices, etc. for the selected target.

The device name syntax described in the PGPLOT manual is also supported; when using this form of device name, the device type is specified using a GNS workstation name.

3.3   Starlink-PGPLOT examples

On non-Starlink systems you may have to replace `/star` by some other path name to locate the files referred to in this section.

The directory /star/share/pgp contains the source of a number of example programs which demonstrate most of the features of PGPLOT. Binaries of the example programs can be found in /star/bin/examples/pgp.

They can be run with a command such as:

```
/star/bin/examples/pgp/pgdemon
```

where n is between 1 and 14, provided that they have been installed. (They may not have been in order to save disk space).

3.4   Plotting in the current viewport

PGPLOT can be used to plot a picture in the current viewport on an already open GKS worksta-
tion. When used in this way, the second argument to PGBEG (normally the workstation name) is
a GKS workstation identifier (encoded as a character string). PGPLOT then behaves as if the
region of the display surface defined by the current viewport is a complete workstation. When
PGEND is called the workstation is not closed but the state of GKS is restored to what it was at the
time that PGBEG was called.

PGPLOT assumes that it has exclusive control over the GKS and so the only graphics calls
allowed between PGBEG and PGEND are PGPLOT routines and GKS inquiry routines.

The following simple example is a subroutine that uses PGPLOT to draw an X, Y plot in an SGS
zone.

```
      SUBROUTINE XYPLOT (IZONE, X, Y, N, XLO, XHI, YLO, YHI, ISTAT)
*++
*
*    XYPLOT   Draw X,Y plot in an SGS zone.
*
*    Description:
*
*       Uses PGPLOT to draw an X,Y plot of the real arrays X & Y in the
*       region of the display surface defined be the specified SGS zone.
*
*    Input arguments:
*
*       IZONE     INTEGER         SGS zone identifier
*       X         REAL(N)         X values of data points
*       Y         REAL(N)         Y   "     "    "     "
*       N         INTEGER         Number of data points
*       XLO       REAL            Lower X axis limit
*       XHI       REAL            Higher X   "     "
*       YLO       REAL            Lower Y   "     "
*       YHI       REAL            Higher Y  "    "
*
*    Output arguments:
*
*       ISTAT     INTEGER         SGS status
*
*    Side effects:
*
*       The specified SGS zone is selected.
*++
      IMPLICIT NONE
      INTEGER  IZONE, N, ISTAT, PGSTAT
      REAL     X(N), Y(N), XLO, XHI, YLO, YHI

      CHARACTER*10 WKID
      INTEGER IWKID

      INTEGER PGBEG
```

```
*  Select the specified SGS zone.
     CALL SGS_SELZ(IZONE, ISTAT)
     IF (ISTAT.EQ.0) THEN

*      Inquire the GKS workstation identifier of the current zone.
         CALL SGS_ICURW(IWKID)

*      Encode workstation id as a character string.
         WRITE(UNIT=WKID, FMT='(I10)') IWKID

*      Open PGPLOT
         PGSTAT = PGBEG(0, WKID, 1, 1)
         IF (PGSTAT.EQ.1) THEN

*        Define axis limits.
             CALL PGENV(XLO, XHI, YLO, YHI, 0, 0)

*        Plot the data.
             CALL PGPT(N, X, Y, 2)

*        Close down PGPLOT.
             CALL PGEND
           END IF
        END IF
        END
```

Because other plotting packages may have plotted on the same physical device, there are some restrictions when using PGPLOT in this way:

- `PGVSIZ` cannot be used.

- `PGPAGE` will never clear the screen. If the display surface has been divided into sub-pictures `PGPAGE` will move to the next sub-picture in the usual way.

- On devices with fixed colour tables, the default PGPLOT colour table will not be set up by `PGBEG` unless the display surface is empty.

4   Other Differences

In general the two versions produce nearly identical results when run on the same device but the following differences should be noted:

- The GKS version of `PGPAGE` cannot be used to make the display surface larger than the default size provided by `PGBEG`.

- The GKS version of `PGBEG` clears the display surface immediately instead of waiting until the first vector is plotted.

5   Support

PGPLOT is Starlink supported software and bugs should be reported through the usual channels and not by contacting Dr Pearson directly. Problems with the GKS specific code will be dealt with by Starlink but all changes to the code which is common to the two versions of PGPLOT must be made in collaboration with Dr Pearson.

The mixing of calls to PGPLOT and GKS routines is not supported except as described in section 3.4, and neither version supports the calling of GRPCKG routines directly. Existing programs that call GRPCKG should be re-written to call the equivalent PGPLOT routines.