Using **pgplot** for graphics

A number of graphics packages are available on the linux boxes. You may use whichever you choose to produce graphs for assignments. However, one that is easy to use and quite powerful is **pgplot**.

Setting up your environment

You will need the following in your .cshrc or .tcshrc file:

```
setenv PGPLOT_PATH /usr/lib/pgplot
setenv PGPLOT_DIR /usr/lib/pgplot
setenv PGPLOT_FONT /usr/lib/pgplot/grfont.dat
set path = ($path /usr/lib/pgplot/)
```

This is then automatically set every time you log in (type `source .cshrc` or `source .tcshrc` to have it read now if you are not going to logout and log back in again).

Linking in pgplot

To compile a fortran program `fred.f` and link in **pgplot**, you could do this:

```
g77 -o fred fred.f -lpgplot -L/usr/X11R6/lib -lX11
```

This compiles with the gnu fortran compiler g77, links in the **pgplot** and X11 libraries, and writes out the exectuable as `fred`.

Running pgplot

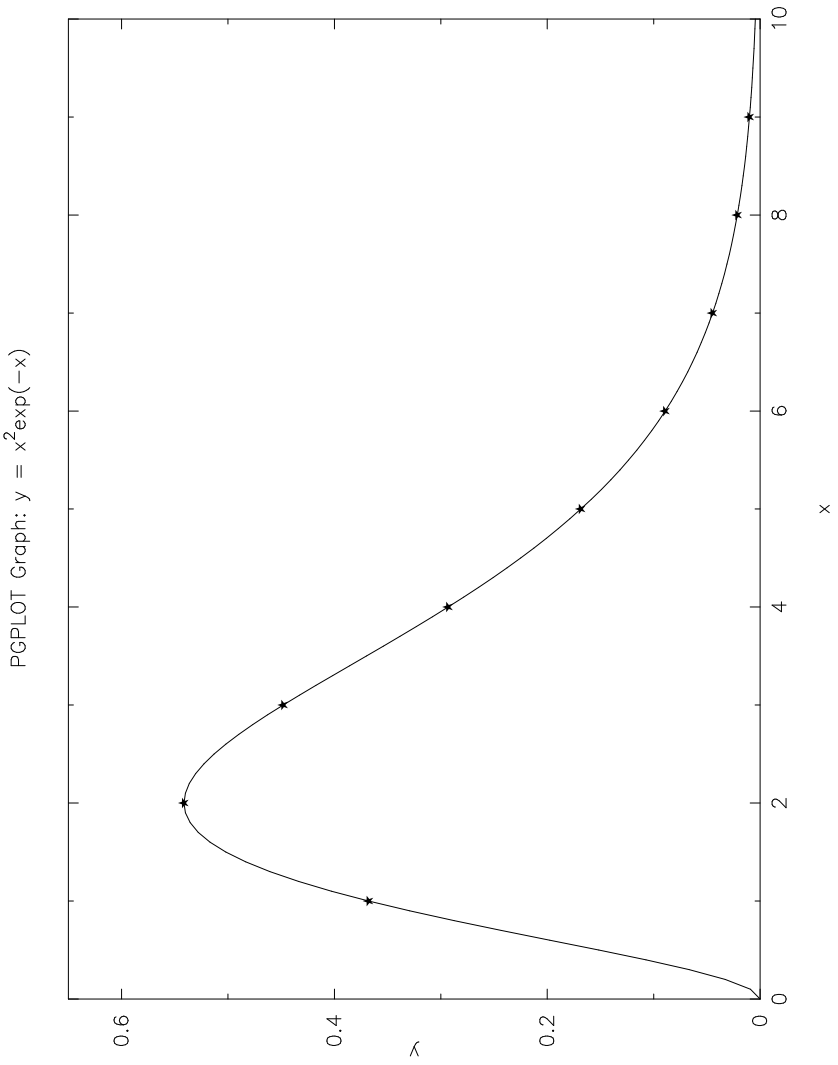
When you run the program it will ask you `Graphics device/type (? to see list, default /NULL):`, to which you should respond `/xw` to get an X window, or `/ps` to send the output to a postscript file `pgplot.ps`. If you choose the latter, you can view the file using `ghostview`, and print individual frames from there if you wish.

Example

Here is a sample program which uses **pgplot**:

```
      PROGRAM EX1
      INTEGER PGOPEN, I
      REAL XS(9), YS(9), XR(101), YR(101)

C     Compute numbers to be plotted.
      DO I=1,101
          XR(I) = 0.1*(I-1)
          YR(I) = XR(I)**2*EXP(-XR(I))
      END DO
      DO I=1,9
          XS(I) = I
          YS(I) = XS(I)**2*EXP(-XS(I))
      END DO

C     Open graphics device.
      IF (PGOPEN('?') .LT. 1) STOP

C     Define coordinate range of graph (0 < x < 10,
C     0 < y < 0.65), and draw axes.
      CALL PGENV(0., 10., 0., 0.65,  0,  0)

C     Label the axes (note use of \u and \d for
C     raising exponent).
      CALL PGLAB('x', 'y', 'PGPLOT Graph: y = x\u2\dexp(-x)')

C     Plot the line graph.
      CALL PGLINE(101, XR, YR)

C     Plot symbols at selected points.
      CALL PGPT(9, XS, YS, 18)

C     Close the graphics device.
      CALL PGCLOS
      END
```

And here is the output graph:



Documentation

You can find complete documentation on **pgplot** at

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