QUICK IDL TUTORIAL: MAKING HARD COPIES OF PLOTS AND 8-BIT IMAGES FOR BLACK/WHITE PRINTERS
January 28, 2002

Carl Heiles

It's one thing to get your plot or image on the screen, and quite another to get a hardcopy. Getting a hardcopy always involves making a postscript file, for which there are many options such as image size and orientation. And one very important option: how many different possible colors? This is important because postscript files that can display many colors are large. You may need this for images, but for simple plots you can get away with just two colors, black and white, and this saves lots of disk space.

You can tackle these issues in two ways. One is to slog through the documentation and write these things yourself. The other is to take advantage of our already having done this.

There are two basic ways to generate a postscript file. One is to read the image pixels directly from the screen and turn them directly into a postscript file. Using our procedures, this is quick and totally painless. However, the output looks ratty for text and graphs, which consist of lines; pixelized lines don't look very good. But you may be willing to put up with this sometimes—if you're in a hurry, or making a hardcopy for your lab notebook, for example. If you want to use this quick and dirty technique but want better-looking results, use a larger window; the pixelization on the hardcopy will be less noticeable.

The second way is to change the output device from your screen to a postscript file, regenerate the plot, and close the file. This takes advantage of postscript's wonderful ability to combine a pixelized image with vector graphics and produces beautiful results. We make this easy, too, but it does involve regenerating the plot or image.

Section 1 below discusses the first method and section 2 the second. They use routines that we've generated; for documentation, use **doc_library** and if you want the files, find them with **which**.

1. QUICK AND DIRTY: WRITING SCREEN PIXELS TO POSTSCRIPT

This consists of two subsections, one for plots and one for images. For plots, we assume grey scale with either two levels (1 bit—black and white) or 256 levels (8 bits—grey scale; some plots have shading). If your color table has fewer than 256 levels, we interpolate it to 256; this is great for grey scale, but if you are using a non-grey colortable it will probably give you weird results.[1]

Both procedures retain the aspect ratio on the window, even if you try to change it with the

[1] If this statement confuses you, see *IDL Tutorial: Color Images, 8-bit and 24-bit.*

keywords. If you want a different aspect ratio, then generate a new window with the desired aspect ratio (using IDL's **window, xsize=256, ysize=512**, for example) or rewrite the procedure for yourself.

1.1. Copying plots with hardplot

We assume you've already generated your plot. Specify the window from which to read the image using IDL's **wset** command. Then type

hardplot

It will ask for the name of the output filename. **hardplot** is a home-grown procedure that has keywords that allow you to various things; the default values are set for reproducing ordinary black/white plots, inverting the white-on-black that you see on your screen to the black-on-white that you should use for a printed output. See the documentation.

Suppose you've called the resulting postscript file *plot.ps.* To view this postscript file before printing—always a good idea—use the UNIX command

xv plot.ps

or, from within IDL, you can invoke this (or any other) UNIX command by putting a dollar sign in front of it:

$xv plot.ps

The *xv* utility has lots of options that you can access by moving the mouse cursor onto the image and clicking the right-hand button. Try it!

Finally, when you're sure you like the result and, also, that you really want to add one more sheet of paper to the world's trash, get a hard copy on the printer with the UNIX command

lp plot.ps

or, from within IDL,

$lp plot.ps

1.2. Copying images with hardimage

This is almost identical to hardplot running with *nbits=8* and the *noreverse* option, but it also copies the colors if the image is not greyscale (which doesn't do you much good with a black/white printer!). We provide the separate home-grown procedure **hardimage** because it's too easy to forget to use the proper options, which wastes lots of printer toner.

2. A BIT LESS QUICK AND REALLY CLEAN

This is a three-step process:

(1) Define the postscript device as the output device with our home-grown **openplotps** (for plots) and **openimageps** (for images);

(2) Generate the plot or image;

(3) Close the ps device and again define the output device as the X-window using our **closeplotps** (same for both plots and images).

openplotps and **openimageps** are basically the same code as used to define the parameters of the ps device in **hardplot** and **hardimage**, and they take almost the same keywords. After you invoke either of these, the ps device is the output device; any **plot** or **tv** commands will write onto the ps file. If you decide in the middle that you've goofed and want to start over again, type **closeplotps** and start over.