G. *PostScript* fonts used by GMT

GMT uses the standard 35 fonts that come with most *PostScript* laserwriters. If your printer does not support some of these fonts, it will automatically substitute the default font (which is usually Courier). The following is a list of the *GMT* fonts:

#	Font Name	#	Font Name
0	Helvetica	**17**	**Bookman–Demi**
1	**Helvetica–Bold**	***18***	***Bookman–DemiItalic***
2	*Helvetica–Oblique*	19	Bookman–Light
3	***Helvetica–BoldOblique***	*20*	*Bookman–LightItalic*
4	Times–Roman	21	Helvetica–Narrow
5	**Times–Bold**	**22**	**Helvetica–Narrow–Bold**
6	*Times–Italic*	*23*	*Helvetica–Narrow–Oblique*
7	***Times–BoldItalic***	***24***	***Helvetica–Narrow–BoldOblique***
8	Courier	25	NewCenturySchlbk–Roman
9	**Courier–Bold**	*26*	*NewCenturySchlbk–Italic*
10	*Courier–Oblique*	**27**	**NewCenturySchlbk–Bold**
11	***Courier–BoldOblique***	***28***	***NewCenturySchlbk–BoldItalic***
12	Σψμβολ (Symbol)	29	Palatino–Roman
13	AvantGarde–Book	*30*	*Palatino–Italic*
14	*AvantGarde–BookOblique*	**31**	**Palatino–Bold**
15	**AvantGarde–Demi**	*32*	***Palatino–BoldItalic***
16	***AvantGarde–DemiOblique***	*33*	*ZapfChancery–MediumItalic*
		34	✺❁❒❄✤❇■✳❂❁▼▲ (ZapfDingbats)

Figure G.1: The standard 35 *PostScript* fonts recognized by **GMT**.

For the special fonts Symbol (12) and ZapfDingbats (34), see the octal charts in Appendix F. When specifying fonts in *GMT*, you can either give the entire font name *or* just the font number listed in this table. To change the fonts used in plotting basemap frames, see the man page for **gmtdefaults**. For direct plotting of text-strings, see the man page for **pstext**. To add additional fonts that you may have purchased or that are available at your institution, see instructions in the CUSTOM_font_info.d under the share/pslib directory.

H. Problems with display of GMT *PostScript*

GMT creates valid (so far as we know) Adobe *PostScript* Level 2. It does not use operators specific to Level 3 and should therefore produce output that should print on all *PostScript* printers[1]. Sometimes unexpected things happen when *GMT* output is sent to certain printers or displays. This section lists some things we have learned from experience, and some work-arounds. Note that many of these lessons are now rather old so hopefully these workarounds no longer apply to anybody...

H.1 *PostScript* driver bugs

When you try to display a *PostScript* file on a device, such as a printer or your screen, then a program called a *PostScript* device driver has to compute which device pixels should receive which colors (black or white in the case of a simple laser printer) in order to display the file. At this stage, certain device-dependent things may happen. These are not limitations of *GMT* or *PostScript*, but of the particular display device. The following bugs are known to us based on our experiences:

1. Early versions of the Sun SPARCprinter software caused linewidth-dependent path displacement. We reported this bug and it has been fixed in newer versions of the software. Try using **psxy** to draw $y = f(x)$ twice, once with a thin pen (`-W`1) and once with a fat pen (`-W`10); if they do not plot on top of each other, you have this kind of bug and need new software. The problem may also show up when you plot a mixture of solid and dashed (or dotted) lines of various pen thickness

2. The first version of the HP Laserjet 4M (prior to Aug–93) had bugs in the driver program. The old one was *PostScript* SIMM, part number C2080-60001; the new one is called *PostScript* SIMM, part number C2080-60002. You need to get this one plugged into your printer if you have an HP LaserJet 4M.

3. Apple Laserwriters with the older versions of Apple's *PostScript* driver will give the error "limitcheck" and fail to plot when they encounter a path exceeding about 1000–1500 points. Try to get a newer driver from Apple, but if you can't do that, set the parameter MAX_L1_PATH to 1000–1500 or even smaller in the file src/pslib_inc.h and recompile *GMT*. The number of points in a *PostScript* path can be arbitrarily large, in principle; *GMT* will only create paths longer than MAX_L1_PATH if the path represents a filled polygon or clipping path. Line-drawings (no fill) will be split so that no segment exceeds MAX_L1_PATH. This means **psxy** `-G` will issue a warning when you plot a polygon with more than MAX_L1_PATH points in it. It is then your responsibility to split the large polygon into several smaller segments. If **pscoast** gives such warnings and the file fails to plot you may have to select one of the lower resolution databases The path limitation exemplified by these Apple printers is what makes the higher-resolution coastlines for **pscoast** non-trivial: such coastlines have to be organized so that fill operations do not generate excessively large paths. Some HP *PostScript* cartridges for the Laserjet III also have trouble with paths exceeding 1500 points; they may successfully print the file, but it can take all night!

4. 8-bit color screen displays (and programs which use only 8-bits, even on 24-bit monitors, such as Sun's ***pageview*** under OpenWindows) may not dither cleverly, and so the color they show you may not resemble the color your *PostScript* file is asking for. Therefore, if you choose colors you like on the screen, you may be surprised to find that your plot looks different on the hardcopy printer or film writer. The only thing you can do is be aware of this, and make some test cases on your hardcopy devices and compare them with the screen, until you get used to this effect. (Each hardcopy device is also a little different, and so you will eventually find that you want to tune your color choices for each device.) The rgb color cube in example 11 may help.

5. Some versions of Sun's OpenWindows program ***pageview*** have only a limited number of colors available; the number can be increased somewhat by starting ***openwin*** with the option "`openwin -cubesize large`".

[1] Note, however, that the `-Q` option in **grdimage** will exercise a *PostScript* Level 3 feature called colormasking.