# PostScript Fonts in LaTeX 2ε

Alan Jeffrey
School of Cognitive and Computing Sciences
University of Sussex
Falmer
Brighton
BN1 9QH
UK
alanje@cogs.susx.ac.uk

**Abstract**

This paper describes the LaTeX 2ε PostScript fonts package psnfss, and the Adobe Times math fonts package mathptm. This paper describes some of the design decisions made in psnfss, and gives an overview of how other fonts can be used in a similar fashion.

## Introduction

This paper describes Sebastian Rahtz's psnfss package for using PostScript fonts in LaTeX. The psnfss software has been the standard way of using PostScript fonts in LaTeX for a number of years, and has recently been updated for LaTeX 2ε and the fonts generated by the author's (1994) fontinst package.

The psnfss package aims to make using PostScript fonts as simple as possible. Once psnfss has been installed, users can select a LaTeX 2ε package such as times and their document will be set in Adobe Times. The psnfss package comes with a standard set of TeX font metric (tfm) files, so psnfss documents should be portable between different TeX implementations.

One of the new features of psnfss is the mathptm package, which allows Adobe Times to be used as a math font as well as a text font. This only uses standard fonts (Adobe Times, Adobe Symbol and Computer Modern) so mathptm documents are portable, and the resulting PostScript files can be distributed without worrying about proprietary fonts.

Together with David Carlisle and Sebastian Rahtz's (1994) graphics and color packages, psnfss will help to free LaTeX from its popular image as only setting academic texts in Computer Modern with `picture` mode graphics.

## Using psnfss

Once psnfss has been installed, it is very simple for users to use. They just select an appropriate package, for example:

```
\documentclass{article}
\usepackage{times}
```

Some document classes will be designed for use with PostScript fonts, and will automatically select PostScript fonts without the user selecting a package. For example, a publishing house producing *The Journal of Dull Results* may have their own jdullres document class, with options for pre-prints or final copy. An author would type:

```
\documentclass[preprint]{jdullres}
```

and would get a pre-print document set in Adobe Times, whereas the production staff would type:

```
\documentclass[crc]{jdullres}
```

to get camera-ready copy set in Autologic Times.

Documents written using psnfss can be printed with an appropriate `dvi` driver such as `dvips`, or OzTeX. Some previewers, such as `xdvi`, cannot preview PostScript fonts without turning the fonts into bitmaps (using up valuable disk space). But the PostScript can be previewed, using `ghostview` or `pageview`.

## PostScript math fonts

One of the common complaints about using LaTeX with PostScript fonts is that the mathematics is still set in Computer Modern, for example as in Figure 2. This is unfortunate, since Computer Modern is a much lighter, wider and more cursive font than suits Adobe Times.

Until recently, the only thing that could be done about this was to use the MathTime fonts, available from Y&Y. These are fine fonts, and can produce excellent math setting. Unfortunately, they are proprietary software, and so cannot be distributed as freely as Computer Modern.

Suppose $f \in \mathcal{S}_n$ and $g(x) = (-1)^{|\alpha|} x^\alpha f(x)$. Then $g \in \mathcal{S}_n$; now $(c)$ implies that $\hat{g} = D_\alpha \hat{f}$ and $P \cdot D_\alpha \hat{f} = P \cdot \hat{g} = (P(D)g)\hat{}$, which is a bounded function, since $P(D)g \in L^1(R^n)$. This proves that $\hat{f} \in \mathcal{S}_n$. If $f_i \to f$ in $\mathcal{S}_n$, then $f_i \to f$ in $L^1(R^n)$. Therefore $\hat{f}_i(t) \to \hat{f}(t)$ for all $t \in R^n$. That $f \to \hat{f}$ is a *continuous* mapping of $\mathcal{S}_n$ into $\mathcal{S}_n$ follows now from the closed graph theorem. *Functional Analysis*, W. Rudin, McGraw–Hill, 1973.

**Figure 1**: Computer Modern text with matching math

Suppose $f \in \mathcal{S}_n$ and $g(x) = (-1)^{|\alpha|} x^\alpha f(x)$. Then $g \in \mathcal{S}_n$; now $(c)$ implies that $\hat{g} = D_\alpha \hat{f}$ and $P \cdot D_\alpha \hat{f} = P \cdot \hat{g} = (P(D)g)\hat{}$, which is a bounded function, since $P(D)g \in L^1(R^n)$. This proves that $\hat{f} \in \mathcal{S}_n$. If $f_i \to f$ in $\mathcal{S}_n$, then $f_i \to f$ in $L^1(R^n)$. Therefore $\hat{f}_i(t) \to \hat{f}(t)$ for all $t \in R^n$. That $f \to \hat{f}$ is a *continuous* mapping of $\mathcal{S}_n$ into $\mathcal{S}_n$ follows now from the closed graph theorem. *Functional Analysis*, W. Rudin, McGraw–Hill, 1973.

**Figure 2**: Adobe Times text with Computer Modern math

Suppose $f \in \mathcal{S}_n$ and $g(x) = (-1)^{|\alpha|} x^\alpha f(x)$. Then $g \in \mathcal{S}_n$; now $(c)$ implies that $\hat{g} = D_\alpha \hat{f}$ and $P \cdot D_\alpha \hat{f} = P \cdot \hat{g} = (P(D)g)\hat{}$, which is a bounded function, since $P(D)g \in L^1(R^n)$. This proves that $\hat{f} \in \mathcal{S}_n$. If $f_i \to f$ in $\mathcal{S}_n$, then $f_i \to f$ in $L^1(R^n)$. Therefore $\hat{f}_i(t) \to \hat{f}(t)$ for all $t \in R^n$. That $f \to \hat{f}$ is a *continuous* mapping of $\mathcal{S}_n$ into $\mathcal{S}_n$ follows now from the closed graph theorem. *Functional Analysis*, W. Rudin, McGraw–Hill, 1973.

**Figure 3**: Adobe Times text with matching math

A less beautiful, but cheaper, solution is to use the mathptm package. This provides drop-in replacements for Computer Modern using virtual fonts built from Adobe Times, Symbol, Zapf Chancery, and Computer Modern. The results can be seen in Figure 3.

The mathptm fonts are distributed free, and the resulting PostScript documents can be made available for anonymous `ftp` without having to worry about unscrupulous readers stealing the fonts from the PostScript documents.

### Roadmap

The rest of this paper describes some technical details about the implementation of psnfss, for the TEXnically minded.

The unpacked psnfss package comes as a number of files:

- Files ending with `sty` are LaTeX $2_\varepsilon$ packages. For example, `times.sty` contains the times package.
- Files ending with `fd` are LaTeX $2_\varepsilon$ font definition files. For example, `T1ptm.fd` contains the font definitions for Adobe Times. This tells LaTeX $2_\varepsilon$ that, for example, Adobe Times bold italic is called `ptmbiq`.
- Files ending with `tfm` are TeX font metric files. For example, `ptmbiq.tfm` contains the font information which TeX needs for Adobe Times bold italic.
- Files ending with `vf` are virtual fonts. For example, `ptmbiq.vf` contains the font information which some printers and previewers need for Adobe Times bold italic. This tells the printer that, for example, the character 'Ć' is made from an 'acute' and an 'C'.

The `sty` and `fd` files are used by LaTeX, the `tfm` files are used by TeX, and the `vf` files are used by printers and previewers.

### Document portability

The psnfss package is intended to make documents as portable as possible. To achieve this, the `sty`, `fd` and `tfm` files should be the same at all sites. This means that documents using the times package will print identically on different sites.

Although the psnfss package distributes all of the files that are used by LaTeX and TeX, it does *not* include the files which are used by particular printer drivers, since these change from site to site. The psnfss package makes no requirements on the printer driver, except that it can print with PostScript fonts.

### Virtual fonts

Although psnfss comes with virtual fonts, these are an optional part of the package. Some printer drivers (such as Y&Y's `dvipsone`) use PostScript font re-encoding rather than virtual fonts. The advantages of virtual fonts include:

- More than one font can be combined together. For example, many PostScript fonts contain 'ff' ligatures in the Expert fonts, but TeX requires ligatures to be in the same font.
- Composite letters such as 'Ă' and 'Ć' can be produced. TeX requires such letters to be in a font for the hyphenation algorithm, but most PostScript fonts do not contain them.
- Most printer drivers and previewers can use virtual fonts, so they are portable between systems.

The advantages of PostScript font re-encoding are:

- PostScript font re-encoding is faster, for example Textures previewing with virtual fonts can be twice as slow as with raw fonts.
- PostScript font re-encoding is a standard technology supported by many other applications.
- In order to access characters like 'Á', the printer driver has to use PostScript font re-encoding anyway, so virtual fonts need two levels of re-encoding rather than one.

Since there is a trade-off between virtual fonts and PostScript font re-encoding, the psnfss package makes no assumptions about using virtual fonts. The `vf` files are available for those who want to use them, but not all sites will want to use them.

### Existing PostScript fonts

In the past, there have been problems with installing psnfss on systems which have already got PostScript fonts generated using Tom Rokicki's `afm2tfm` converter.

The main difference between the fontinst fonts and the `afm2tfm` fonts is that the fontinst fonts are designed to be drop-in replacements for Computer Modern, and can be used with no new macros. The `afm2tfm` fonts contain some new characters (such as '©') and some old characters (such as '") in different slots, so need special macros. See Tables 2 and 3.

One of the most important features of LaTeX 2ε is that different fonts can be used without new macros, which is one of the reasons for using the fontinst fonts rather than the `afm2tfm` fonts.

In the past, there were problems using psnfss on systems which had already got the `afm2tfm` fonts, because they used the same font names. This has now been changed, and fontinst uses the letters `7t` to indicate a '7-bit TeX text' font, and `0` to indicate a 'Adobe Standard' font. For example the filenames for Adobe Times are:

| *encoding* | *fontinst name* | *afm2tfm name* |
|---|---|---|
| Adobe Standard | `ptmr0` | `rptmr` |
| 7-bit TeX text | `ptmr7t` | `ptmr` |
| 8-bit TeX text | `ptmrq` | none |

There should now be no clashes between the fonts generated by `afm2tfm` and those generated by fontinst.

### Font naming

The psnfss fonts are named using Karl Berry's naming scheme. This tries to fit as many font names as possible into the eight letters provided by some operating systems. For most fonts here, the names are given by:

- One letter for the font foundry, e.g. `p` for Adobe.
- Two letters for the font family, e.g. `tm` for Times Roman, `hv` for Helvetica or `cr` for Courier.
- One letter for the weight, e.g. `r` for regular or `b` for bold.
- An optional letter for the shape, e.g. `i` for italic, `o` for oblique, or `c` for small caps. No letter means 'upright'.
- One or two letters for the encoding, e.g. `q` for 'Cork' encoding, `0` for 'Adobe Standard' encoding or `7t` for Knuth's 'TeX text' encoding.

For example:

- `ptmr7t` is Adobe, Times Roman, regular weight, upright shape, TeX text encoding.
- `phvbcq` is Adobe, Helvetica, bold weight, small caps shape, Cork encoding.
- `pcrro0` is Adobe, Courier, regular weight, oblique shape, Adobe Standard encoding. This is the font Adobe call 'Courier-Oblique'.

The naming scheme is described in more detail by Berry (1994). It is far from ideal, but does allow most fonts to be named in a consistent fashion. Most systems require a translation from the TeX font name to whatever the local font name is. For example, `dvips` can be told that `pcrr0` is Adobe Courier with a line in the `psfonts.map` file:

```
pcrr0 Courier
```

OzTeX requires a line in the `Default` configuration file:

```
= pcrr0 Courier Courier Mac.enc
```

Textures can add new font names using the EdMetrics application.

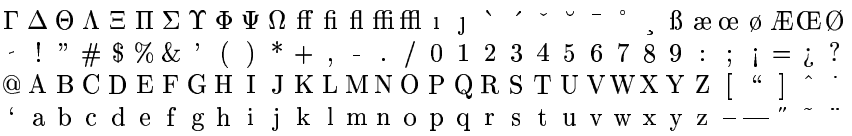

Table 1: Computer Modern Roman

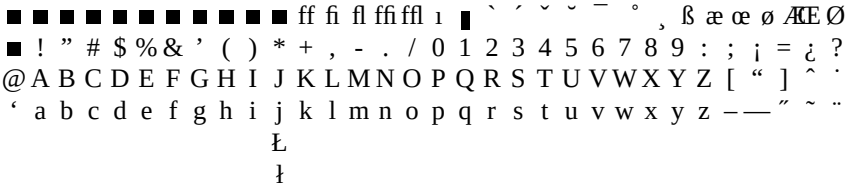

Table 2: The fontinst Adobe Times roman

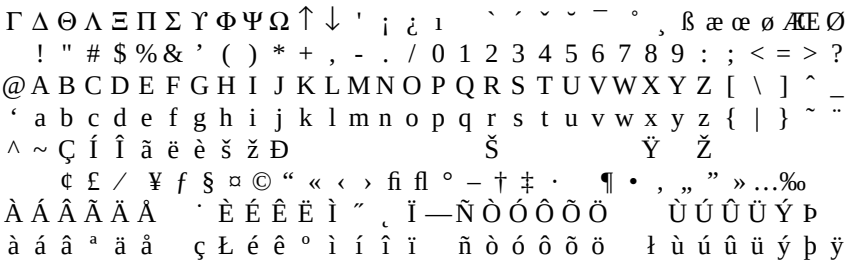

Table 3: The `afm2tfm` Adobe Times roman

### The mathptm fonts

The mathptm fonts are an interesting exercise in using the fontinst package to generate quite complex virtual fonts.

The mathptm fonts are shown in Tables 4–7:

- `zptmcmr` is the 'operators' font, used for operators such as `\log` and `\sin`. It is built from Adobe Times roman (for most letters), Adobe Symbol (for upper-case Greek), and Computer Modern roman (for some symbols such as =).
- `zptmcmrm` is the 'letters' font, used for math italic. It is built from Adobe Times italic (for most letters), Computer Modern math italic (for most symbols) and Adobe Symbol (for Greek and `\wp`). The Adobe Times letters have their sidebearings changed for mathematics.
- `zpzccmry` is the 'symbols' font, used for math symbols. It is built from Computer Modern symbols (for geometric symbols such as `\oplus`), Adobe Symbol (for humanist symbols such as `\nabla`), Adobe Times (for text symbols such as `\P`) and Adobe Zapf Chancery (for calligraphic upper-case).
- `zpsycmrv` is the 'big symbols' font, used for large symbols such as '$\sum$'. It is built from Computer Modern extensions (for most symbols) and Adobe Symbol (for `\sum` and `\prod`). The big operators such as `\bigcup` are set in 9pt rather than 10pt since these blend with the smaller operators from Adobe Symbol.

Since the virtual fonts are built from commonly available fonts, rather than specifically designed fonts such as MathTime, there are some oddities. The parentheses are from Computer Modern rather than Adobe Times, since they have to blend with the extensible parentheses from `zpsycmrv`; this means the math parentheses are different from the text parentheses. Adobe Symbol has no large '$\sum$' and '$\prod$', so these have to be faked by magnifying the text characters, which does not look great. Some glyphs, such as `\coprod` are missing. The 'letters' Greek is upright rather than italic.

But despite these features, the fonts are acceptable, and make TEXnical typesetting with generally available scalable fonts possible for the first time.

### Installing your own PostScript fonts

The psnfss fonts were all built using the fontinst font installation package. This package is written entirely in TEX, so it can be run on any TEX installation with enough memory. It is quite slow (about 20 minutes for a font family on a Macintosh with a

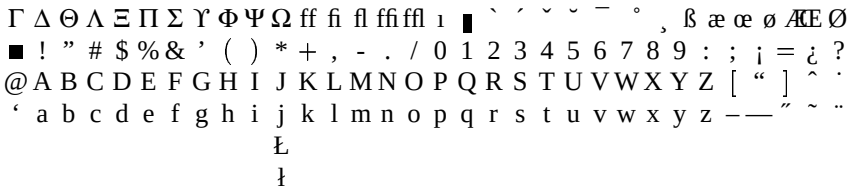

Table 4: The mathptm 'operators' font

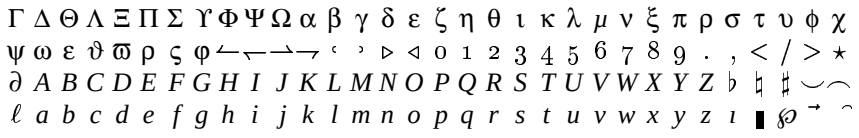

Table 5: The mathptm 'letters' font

Table 6: The mathptm 'symbols' font

Table 7: The mathptm 'big symbols' font

33MHz 68030) but this is acceptable since it is not run often.

Version 0.19 was described by the author (1993), but the user interface has changed considerably since then.

To install a new font using fontinst, you need the Adobe Font Metrics (`afm`) files for the fonts. These should be named with Karl Berry's naming scheme, for example `ptmr0.afm` for Adobe Times Roman. They should be in a directory which can be read by TEX.

Installing Latin fonts is quite simple. For example, to install the Adobe Times fonts, you run TEX on `fontinst.sty` and say:

```
\latinfamily{ptm}{}
```

If you have bought the Expert fonts, you can install Adobe Times Expert and Adobe Times Old Style by saying:

```
\latinfamily{ptmx}{}
\latinfamily{ptm9}{}
```

Once TEX has finished, it produces:

- `pl` files, which are text representations of `tfm` files. They can be converted to `tfm` files with `pltotf` (part of EdMetrics in Textures and OzTools in OzTEX).
- `vpl` files, which are text representations of `vf` files. They can be converted to `vf` files with `vpltovf` (part of EdMetrics in Textures and OzTools in OzTEX).
- `fd` files, which are used by LATEX $2_\varepsilon$.

The Adobe Times fonts can then be used in LATEX by saying:

```
\renewcommand{\rmdefault}{ptm}
\rmfamily
```

If you want to write your own package similar to times, you just create a `sty` file containing these lines. For example, an Adobe Times Old Style package would contain:

```
\ProvidesPackage{timesold}
   [1995/04/01 Times old style digits]
\renewcommand{\rmdefault}{ptm9}
\rmfamily
```

It is also possible to create customized fonts (such as the mathptm math fonts) using fontinst, but this is somewhat trickier, and is described in the fontinst documentation.

If you use fontinst to install some fonts, and you would like to distribute the results, please mail them to me, and I'll include them in the fontinst distribution.

## Ongoing work

There are a number of areas of ongoing work with fonts and TEX.

There is a TUG working group on TEX directory structures, which will recommend how fonts should be installed on any TEX system. This will make it easier to distribute TEX software, because there will be a standard directory structure to refer to. The CTAN `fonts/metrics` directory will reflect the TDS structure.

The fonts produced by fontinst do not include 'eth' or 'thorn' because these characters are not available in Adobe Standard encoding. We are working on developing a suitable replacement for Adobe Standard encoding which will allow access to 'eth', 'thorn' and the other missing characters.

There is a TUG/LATEX working group on math font encodings, which will develop symbol encodings which will be supported by fontinst (this working group has been rather slowed down by the production of LATEX $2_\varepsilon$).

## Bibliography


Berry, Karl. *Filenames for fonts*, available as `fontname.texi` from CTAN, 1994.

Carlisle, David and Rahtz, Sebastian. *The color and graphics LATEX $2_\varepsilon$ packages*, available from CTAN, 1994.

Goossens, Mittelbach and Samarin, *The LATEX companion*, Addison-Wesley, 1993.

Jeffrey, Alan. A PostScript font installation package written in TEX, in *Proc. 1993 TUG AGM.*

Jeffrey, Alan. *The fontinst package*, available from CTAN, 1994.

Rahtz, Sebastian. *Notes on setup of PostScript fonts for LATEX $2_\varepsilon$*, part of the psnfss package, available from CTAN, 1994.