

The elementary way to use fonts in PostScript

You will very often want to put text in your figures. PostScript's font-handling capabilities are very good, but most of the techniques are designed to be automated by some other program, because good text, especially mathematical text, requires a lot of work to get font choices, spacing, and sizes right.

In this note I will explain how to place simple text in PostScript figures. What I explain there will not make it easy to do sophisticated typesetting of labels in figures, but it will be adequate for many purposes.

Simple PostScript text

The simplest way is just to use the generally available PostScript fonts to assemble your text 'by hand', i.e. by thinking out for yourself what layout, font choice, and text size are to be. This is a relatively straightforward process, and probably your best choice, if the text you want to include is not too complicated. The basic principle is that PostScript treats letters as paths.

There are three steps you must carry out each time you want to use a new font.

(1) You must decide which font you are going to use. There is a limited choice of fonts guaranteed to be available on all implementations. The choice is, roughly, from this list: Times-Roman, Times-Italic, Times-Bold, Times-BoldItalic, Helvetica, Helvetica-Oblique, Helvetica-Bold, Palatino, Courier, Courier-Bold, Palatino, Palatino-Bold, Symbol. Of course, you can switch back and forth among several fonts. You can use other PostScript fonts, too, but you ought to include them explicitly in your file.

The Helvetica fonts have no serifs, and display well on a computer. The Symbol font contains Greek letters.

(2) You decide what scale you are going to use it at. With the command `scalefont`, you set the vertical size of the letters in terms of the current unit.

(3) You apply the command `setfont`.

If the current unit is one inch, for example, this will give you letters 1/4" high:

```
/Helvetica-Bold findfont
0.25 scalefont
setfont
```

To put text on a page, you move to the spot you want the text to begin, and then use the `show` command. The text itself is enclosed in parentheses.

```
0 0 moveto
(Geometry) show
```

after the previous command sequence will give you

Geometry

To print out the value of a variable, you must convert the variable to a string with the command `cvs`, using an empty string of sufficient size to hold the variable's value.

```
0 0 moveto
(x = ) show
x (      ) cvs show
```

will produce this, if $x = 3$:

x = 3

Scaling of a font is special to the font, but scaling the figure will affect the font also. So will general coordinate changes. You can get interesting effects.

Geometry
γιτəmoəθ

By choosing the right affine transformation, you can simulate fonts written in 3D and projected onto the page.