Where Are the Math Fonts?

Berthold K.P. Horn
Y&Y
106 Indian Hill
Carlisle
MA 01741
USA
Internet: bkph@ai.mit.edu

Abstract

Everyone knows that there are very many choices for text fonts for use with TEX, including over 14,000 (*fourteen-thousand!*) fonts in industry standard Adobe Type 1 format, plus several hundred in other common formats such as TrueType. There are, however, relatively few fonts with mathematical symbols, operators, delimiters, and relations. And *very* few of these can be used with TEX.

Why So Few?

Right now, there are few basic math font sets for TEX beyond the following four:

- Computer Modern math fonts;
- Lucida Math;
- Lucida New Math; and
- *MathTime*

One reason there are so few is that there are relatively few 'math fonts' to start with. But *much* more importantly, a 'math font' — as far as TEX is concerned — is much more than a mere collection of glyphs, and furthermore, TEX imposes severe and peculiar constraints on those glyphs. Hence, to be useful with TEX, a math font set has to be explicitly designed for TEX. In addition, tailoring a math font set for use with TEX means that it will most likely not be very useful for anything but TEX. This greatly reduces the incentive for putting in the enormous work required to create and develop a new math font set.

What Are the Special Requirements that TEX Imposes?

The requirement that is least restrictive, and easiest to explain, is that TEX requires metric files in its own particular compact binary format. In the case of text files, such TEX metric files are quite easy to create, containing primarily character advance width, kerning and ligature information. Tools are available for creating TEX metric files automatically from other formats, such as the human readable Adobe font metric format.

But TEX metric files for *math* fonts must contain a lot more. This includes information for each letter on how to position subscripts and superscripts, and also how to place accents. Furthermore, in the case of the math extension font, a complex bit of machinery is needed to link together delimiters of the same basic shape but different size, and to describe how even larger delimiters can be constructed by splicing together partial glyphs. Additional 'font dimensions' must also be specified giving information on where the 'math axis' is, how to place numerator upon denominator, and so on.

But generating appropriate `tfm` files is actually a very small part of the problem.

Constraints on Math Fonts Used with TEX

First of all, a math font must contain information on how to properly position subscripts and superscripts. This is done using character width and the so-called 'italic corrections'. The subscript is placed at a position determined by the character 'width', while the superscript is placed at a position determined by the sum of the character 'width' and the 'italic correction'. Note that this means that the stated character 'width' is *not* the overall desired advance width for that character at all — instead the advance width is the character 'width' plus the 'italic correction'!

This has additional consequences. Normally TEX uses the difference between the characters 'height' and the stated x-height for the font to adjust the vertical position of accents. TEX uses the character and the accent's widths to center the accent horizontally over the character. Since in the case of math fonts, the stated 'width' of the character is in fact *not* the advance width, TEX's normal calculation of accent positions no longer works. To compensate, fake 'kern pairs' are introduced — involving a specified 'skew character.' These do not specify kerning at all, but instead specify the position of an accent in math mode. So TEX math fonts must use basic metric information such as character width and pair kerning information in non-standard ways. Clearly use of such a font with applications other than TEX will be seriously impacted by this.

Next, large delimiters 'hang off the baseline' rather than being centered on the math-axis, for example. That is, the character 'height' above the baseline is very small, or even zero. This means that these delimiters are useless for anything but TEX. The same goes for large operators, radicals, and integrals. Consequently, a typical 'math extension' font is something only useful for TEX.

Which brings us to leading. Most applications compute suitable spacing between lines based on the ascenders and descenders in a font in order to avoid glyphs from adjacent lines bumping into each other. This works fine for a typical text font with capheight around 0.75 of an em, and descender around 0.25 of an em. It clearly will not work as desired if a line contains even a single character from a math extension font, since this might have a descender between 2 and 3 times an em. But then we already decided that a math extension font is 'TEX-specific'. Unfortunately, the same problem applies to a 'math symbol' font, at least if one sticks to anything like the layout of characters using in the CM math fonts.

The reason is that TEX uses the character 'height' of the 'radical' character as the thickness of the horizontal stroke of a radical. So a radical in a normal text position would induce an extremely thick top bar on a square root! So, once again, the 'radical' symbol has to 'hang off the baseline.' This single glyph then greatly increases the descender of the math symbol font and makes it hard to use with anything but TEX.

TEX's algorithms for laying out mathematical formulæ are truly wonderful and truly complex. They also contain hard-wired constants and hard-wired assumptions. These assumption are all reasonable, of course, for Computer Modern fonts, but may not be appropriate for other fonts. For example, it is assumed that the 'math axis' is also the 'delimiter axis'. That is, that the vertical center of mathematical operators falls at the same level as the vertical center of the normal size delimiters.

Now, some of the very features described above as problematic are ones that contribute to TEX's superb capabilities in typesetting mathematical material. So we couldn't do without them. What *is* unfortunate is that these require fundamental changes to the font itself — rather than just the TEX metric files — for a math font to be useful with TEX. We would be able to use many more of the existing math fonts with TEX if it was just a matter of adding extra trickery to the TEX metric file! There are already programs that can create `tfm` files from `afm` files for math fonts, but they only work for fonts that have been to designed from the ground up with TEX's very special requirements in mind.

Other Peculiarities of Fonts for TEX

Fonts designed for use with TEX have some other features that make them hard to use with anything else. First of all, they use the control character range (0-31), which is not accessible with other applications, since control characters are used for other purposes. Special tricks have to be used to work around this.

Next, fonts designed for TEX do not have a 'space' character in character code position 32, mostly because TEX uses a clever method for deciding how large a space is really needed. This is also a serious handicap. Imagine trying to create illustrations and matching the nomenclature with the text. If the text uses fonts designed for use with TEX then the fonts won't have a 'space' character. It is not that uncommon, however, for captions to require spaces.

There are many other less obvious problems like this. For example, the math symbol font has two zero width characters ('mapsto' and 'negationslash'). Now in most font metric formats, zero width in the metrics means there is no character in that position. In fact, this is even true of the TEX metric format. To quote the bible:

> The `width_index` should never be zero unless the character does not exist in the font, since a character is valid if and only if it lies between `bc` and `ec` and has a nonzero `width_index`.

TEX metric files do not represent widths directly, instead they use an index in a width table, and while the zero-th entry in the table *is* supposed to be zero width, other entries may also be, and so can be used to get around the problem.

Clearly, designing fonts to work well with TEX means they may not be easily useable with other applications — which seriously curtails any interest a font designer might have in such a project.

Some problems can be 'solved' using virtual fonts, but again, virtual fonts are unique to TEX. If a font is to be used both in text and in included drawings produced using arbitrary drawing applications, then 'real' fonts have to be created for the purpose.

Customer Support Questions

When a foundry sells a text font set, there is very little needed in the way of installation instructions or customer support. Text fonts generally are laid out the same way, and installed the same way. Few technical question arise, and there is no need for auxiliary files to 'support' use of the fonts. Customer calls typically have to do with such trivial matters as receiving bad diskettes, or fonts being for the wrong platform.

Not so with math font sets for TEX! Aside from TEX metric files, it is expected that the vendor supply TEX macro files that make it easy to 'switch' to the new font set (the assumption being that one always *starts* with Computer Modern). There is also a need for information on how to create new TEX 'formats' that use the new fonts. And lots of explanatory material in case there are any differences in layout with respect to the way Computer Modern happens to work. Typically the support files require more space than the fonts themselves, and the documentation is substantial.

Customer support can be a serious drain on resources. Much of this is end-user education, since literature about TEX is almost totally focused on use of bitmapped Computer Modern fonts, and some still find it hard to accept that (a) TEX can be used with fonts other than Computer Modern, (b) TEX can be used with fonts that are not in pk bitmapped form, (c) Computer Modern fonts are available in formats other than bitmapped pk files. And the vendor needs to be ready to forever explain why a math font set is not *exactly* like the Computer Modern math font set.

All of this is made more difficult by total lack of standardization of DVI processors in the important areas, such as font encoding and font naming. (We won't even mention figure inclusion!) A great deal of the auxiliary information that has to be provided is there because different drivers require different types of 'configuration' information, and some even use their own unique formats for the basic metric information. In addition, the capabilities of DVI drivers to deal with fonts in scalable outline form (some force the user to resort to virtual fonts), and the abilities to reencode fonts to a user specified encoding, are often limited, and typically not properly documented.

Conclusions

The market for fonts in general is huge, but the market for TEX fonts is tiny. While MicroSoft has already sold several *million* copies of their first TrueType font pack, the market for TEX-specific fonts at the moment is probably only in the *thousands*. Development costs for fonts that are *not* TEX-specific can be spread over a thousand times as many users! Ideally then, TEX should be able to easily use fonts in all sorts of formats developed for other purposes. Conversely, fonts developed for use with TEX should be usable with other applications.

The reason we do not see use of a much wider variety of fonts in TEX, is that fonts used for text and math should harmonize, hence the number of choices is really restricted by the number of 'math fonts' available for use with TEX. So the limit on the number of math fonts that work with TEX is a serious obstacle to the use of a wider variety of fonts.

If we become more flexible in what we have TEX do, then we can latch onto the express train of development of font technology — if, on the other hand, we refuse to acknowledge there are useful ideas outside the TEX world, then we will miss it.