Observations on the implementation of math extension fonts through virtual fonts

Ulrik Vieth

1 Introduction

TEX's extremely sophisticated algorithms for typesetting math formulae make a number of explicit and implicit assumptions about the way math fonts are designed. While some parameters governing the placement of elements in math formulae are specified in the `\fontdimen` parameters, some others are assumed to be hidden in the dimensions of individual characters when used in a certain context.

This paper summarizes some lessons learned concerning some difficult technical points in the implementation of math fonts through virtual fonts, which are easily overlooked when just inspecting font tables and which may require special attention and fine tuning for optimal results.

This paper was inspired by the author's work on an implementation of the old TEX encodings using the Mathematica 3.0 symbol fonts, which proved to be a little troublesome for various reasons. It also incorporates some considerations that were brought up in Berthold Horn's paper *Where are the math fonts?* in the TUG '93 proceedings (*TUGboat* 14#3, pp. 282–284).

2 Technical considerations

2.1 Accents

TEX assumes that the normal and wide math accents are suitably positioned for use with lowercase letters that do not exceed *x_height*. For capital letters and lowercase letters with ascenders, the accents are automatically shifted upwards by the difference of the actual height of the character and *x_height* (Rule 12 in Appendix G of *The TEXbook*).

If a base font provides raw accent characters that are positioned on the baseline, as it is the case in some of the Mathematica symbol fonts, it is necessary to adjust their positioning in the virtual font by shifting them upwards by at least the amount of *x_height* and an appropriate amount of clearance. It is also necessary to ensure that the *x_height* parameter of the fonts containing the accent glyphs is consistent with that of the math letters font.

As for which amount of clearance is appropriate, one may take as a guideline TEX's built-in rules for the placement of `\overline` atoms, in which case the amount of clearance is taken to be three times the default *rule_thickness* (Rule 9 in Appendix G). For optimal results, it seems best to adjust the placement of the `\bar` accent, so that it will be consistent with the `\overline` placement. Once the placement of the `\bar` accent is determined, one may adopt this position for the remaining math accents as well.

2.2 Radicals

TEX assumes that the glyphs containing radical signs are designed in a very special way. The height of the radical sign is assumed to be the same as the height of the horizontal rule to be put on top of the root (Rule 11 in Appendix G) with the consequence that the remainder of the radical sign will have to end up below the baseline.

This assumption has a number of rather unfortunate consequences and also implies that a number of criteria have to be fulfilled in math symbol fonts providing radical signs.

(1) Since the major portion of the radical sign necessarily has to lie below the baseline, the characters will most likely be unsuitable for direct use with less sophisticated typesetting systems than TEX. If a font is to be designed for use with other systems as well, it may be necessary to supply another set of centered radical signs in separate font table slots. An example of such a font is the Mathematica symbol font `Math2`, which includes a special set of radical signs suitable for use with TEX in addition to those intended for use with Mathematica.

(2) The height of the radical sign should match exactly the height of its horizontal rule part (if it is included as part of the character, that is), or else it should be consistent with the default *rule_thickness*. In the Computer Modern fonts, all the radical glyphs have a height of *rule_thickness* anyway, so it seems doubtful whether there is any advantage at all in having the rule thickness used in root signs depend on the character dimensions rather than always using the default *rule_thickness* taken from one of the `\fontdimen` parameters.

(3) There should be no whitespace at the top of the radical sign within the character dimensions since that amount would also be added to the rule thickness used. If there is any extra whitespace, it is necessary to cancel it out by adjusting the TFM height of the radical sign in the virtual font. An appropriate amount of clearance above the radical sign is added automatically by TEX's math typesetting algorithms, so there is no need to do so in the glyph metrics.

(4) There should be no whitespace at the top or bottom of the pieces used to construct extensible radicals. Otherwise, ugly gaps in the extensible radical signs may ensue. (This applies to extensible delimiters as well.) If there is any extra whitespace, it is necessary to cancel it out as explained above.

Apart from the top piece of the radical, it does not matter how the heights and depths of the individual pieces are distributed. However, it may be advantageous to use the same height and depth for all three pieces for other reasons.

2.3 Delimiters

TeX automatically centers big delimiters on the math axis with respect to the total of their height and depth. Thus, it may well be possible to place the delimiters in a centered position, suitable for use with other typesetting systems as well as with TeX. If the font is to be used exclusively with TeX, however, there might be a slight advantage in placing the big delimiters in a lowered position, since this allows to use exactly the same sizes as for the radicals, thereby reducing the number of different TFM heights and depths (which are both restricted to no more than 15 different values each).

For the extensible big delimiters, the individual pieces are stacked on top of each and the whole construction is centered with respect to the resulting total height. It does not matter how the heights and depths of the individual pieces are distributed, but it is important that the characters are designed to be stacked on top of each other without any gaps of intervening whitespace.

The math axis on which the big delimiters are centered is the same as the math axis used for centering big operators, fractions, and `\vcenter` atoms. Should the design of a certain font call for an asymmetric placement on an independent delimiter axis, such an effect will be difficult to achieve within the context of the present TeX, except perhaps by deliberately manipulating the character dimensions. However, it seems plausible that it may be easily implemented within the scope of ε-TeX or $\mathcal{NTS}$.

2.4 Operators

As in the case of big delimiters, TeX also centers big operators on the math axis with respect to the total of their height and depth. Thus, it may well be possible to place big operators on the baseline or in a centered position, suitable for use with other typesetting systems as well as with TeX. Since TeX only makes a distinction between either `\displaystyle` or `\textstyle` and below, there is no advantage in providing more than two sizes of big operators linked by `charlists`, although other systems might use the additional sizes, if available.

If the limits are to be placed above and below the big operators, the spacing is controlled by five `\fontdimen` parameters in the math extension font (Rule 13 of Appendix G). Thus, there is no need to include an appropriate amount of clearance as part of the big operator glyphs. In fact, it may even be preferable to have no vertical whitespace within the glyph's character box at all to ensure symmetric placement of upper and lower limits.

If the limits are to be placed as subscripts and superscripts to the right of the operator symbol, the usual rules for ordinary math characters apply, i. e. the TFM advance width contains the the position of lower limits while the italic correction contains the offset between the position of lower and upper limits (typically used in slanted integrals).

2.5 Horizontal braces

While TeX's `charlist` mechanism allows to link both vertical and horizontal objects of increasing size, such as big delimiters of different height or wide accents of different width, there is no horizontal counterpart of vertical extensible delimiters. Horizontal extensible delimiters such as over- and underbraces therefore have to be constructed using individual pieces for the ends and the middle and an `\hrulefill` to fill the gaps.

As a particular complication, the present OMX encoding uses the horizontal brace tips for a dual purpose to construct the middle piece form two end pieces, which implies that all four pieces must be constructed to have the same TFM height and depth, so that the horizontal parts line up when placed next to each others. In fact, the actual TFM height is just the height of the horizontal rule, while the height or depth of the brace tips extending above or below the rule do not show up in the metrics.

If the horizontal braces in OMX are constructed from a base font containing pieces for over- and underbraces in a different position, it is necessary to shift the characters and adjust their TFM heights and depths in a virtual font implementation of the OMX encoding to make them just barely usable. This problem will hopefully disappear in the new MX encoding where dedicated slots for the middle pieces and the horizontal extension pieces are provided, so that it will no longer be necessary to bother about using characters for a dual purpose.

Still the new MXP/MX1 encodings do not plan to provide slots for ready-made versions of horizontal braces in four different widths to be addressed via `charlists` similar to wide accents, nor does it take into account the possibility of horizontal over/under parentheses, brackets or angle brackets.

◇ Ulrik Vieth
Heinrich-Heine-Universität
Institut für Theoretische Physik II
Universitätsstraße 1
D-40225 Düsseldorf, Germany
`vieth@thphy.uni-duesseldorf.de`