Understanding the æsthetics of math typesetting

Ulrik Vieth
Vaihinger Straße 69
70567 Stuttgart
Germany
ulrik dot vieth (at) arcor dot de

Abstract

One of the core strengths of TEX is the ability to typeset math to very high level of æsthetic standards. However, this level of quality not only depends on TEX alone, but relies on close interaction between sophisticated algorithms (built into the TEX engine) and the fine-tuning of metric information (built into math fonts), which is not so well understood.

At a previous conference Bogusław Jackowski presented a paper *Appendix G Illuminated*, in which he translated the formal description of TEX's algorithms for math typesetting into a visual representation, illustrating the mathematical and geometric relations between the various font metric parameters. While this helps to improve the understanding, it doesn't resolve the question how to determine good values of font metric parameters when designing a new math font.

In this paper, we analyze the values of these parameters in existing fonts and draw some conclusions about the underlying design principles. In the end, we hope to obtain a recipe how to determine good values of font metric parameters based on simple design parameters such as the x-height or rule thickness.

1 Introduction

Math typesetting certainly counts as an application where TEX is best known for its high level of æsthetic quality, perhaps even so much that it is sometimes taken for granted. Nevertheless, the quality of math typesetting is a non-trivial subject, as it crucially depends on the close interaction between algorithms (built into the TEX engine) and metric parameters (built into math fonts).

While the algorithms determine how things are done when the elements of a math formula are assembled, the font metric parameters determine how much the various elements are shifted or adjusted. Thus, the overall quality ultimately depends much more on the setup of the math fonts than on the TEX engine itself.

Moreover, while the algorithms (provided by the TEX implementation) can be considered a fixed reference point, the metric parameters (provided by font designers) have to be revisited every time a new family of math fonts is set up for use with TEX. Thus, a good understanding of these parameters and their values remains important, especially when implementing new math fonts while moving towards Unicode and OpenType font technology.

In principle, TEX's algorithms for typesetting math formulas are well documented, as Appendix G of *The TEXbook* [1] devotes a whole chapter on this topic. However, while this description may be very precise and clear in algorithmic and mathematical terms, it may be hard to follow for a font designer used to thinking in geometric terms.

At a previous conference Bogusław Jackowski presented a paper *Appendix G Illuminated* [3, 4], in which he translated the formal description of TEX's algorithms into a visual representation, illustrating the mathematical and geometric relations between the various font metric parameters.

In the following, we will use his paper as a starting point for further discussions of the font metric parameters, trying to obtain a recipe how to determine good values of font metric parameters when setting up new math fonts.

2 Understanding font metric parameters

In traditional TEX engines, fonts are represented by their font metrics in TFM files.[1] Besides the glyph metrics of individual glyphs[2] as well as ligature and kerning tables, these TFM files also contain a global table of font metric parameters.

[1] This may be subject to change in new TEX engines such as XƎTEX or LuaTEX using OpenType fonts directly.

[2] The glyph metrics of math fonts may be required to have some peculiar properties, see [5, 6] for further discussion.

σ_5	x-height*	σ_{18}	superscript drop
σ_8	numerator 1	σ_{19}	subscript drop
σ_9	numerator 2	σ_{20}	delimiter 1
σ_{10}	numerator 3	σ_{21}	delimiter 2
σ_{11}	denominator 1	σ_{22}	math axis*
σ_{12}	denominator 2	ξ_8	rule thickness*
σ_{13}	superscript 1	ξ_9	big operator 1
σ_{14}	superscript 2	ξ_{10}	big operator 2
σ_{15}	superscript 3	ξ_{11}	big operator 3
σ_{16}	subscript 1	ξ_{12}	big operator 4
σ_{17}	subscript 2	ξ_{13}	big operator 5

Table 1: Summary of font metric parameters related to math typesetting following the notation of Appendix G. Parameters marked with * are assumed to be fixed by the font designer. The remaining parameters have to be determined and expressed in terms of other parameters.

A standard text font is expected to have seven font metric parameters, containing information such as the font slant, the x-height, the quad width, or the interword and extra space.

In addition to that, the math fonts assigned to family 2 and 3 (usually representing the symbol and extension font) are required to have some additional parameters (22 in family 2 and 13 in family 3), which are traditionally denoted as σ_1 to σ_{22} and ξ_1 to ξ_{13}, following the notation of Appendix G.

The full list of font metric parameters related to math typesetting is summarized in Table 1. Some of these parameters, such as the x-height (σ_5), the math axis (σ_{22}), or the rule thickness (ξ_8) are easy to understand and straight-forward to define. Most of the remaining parameters, however, are rather complicated and often remain a mystery to a font designer without further explanations.

In the following, we try to develop a recipe how to determine the values of the remaining parameters in terms of the most basic parameters.

3 Understanding basic design principles

How can we go about to determine the values of font metric parameters? We can start by analyzing the values of these parameters in existing math fonts and we can try to draw some conclusions about what might be the underlying design principles.

In addition, we can consult the sources of the Computer Modern fonts [2] that also contain some calculations of the parameter values, which we can try to understand in order to discover a recipe that can be adapted to other fonts.

To start gaining a better understanding of the underlying principles, it is interesting to study the boundary cases where certain values are always used regardless of the parameter settings.

For example, when typesetting an overline or underline, TeX always draws a rule of thickness θ and applies an inside clearance of 3θ (between the nucleus and the rule) and an outside clearance of θ (above or below the rule), where $\theta = \xi_8$ is the default rule thickness.

A similar principle is used in boundary cases when typesetting fractions when it is necessary to prevent collisions. In such cases, TeX draws a rule of thickness θ and applies a minimum clearance above and below the rule, which is set to either 3θ for the bigger sizes (in display style) or θ for the smaller sizes (in text style or below).

Yet another similar, but slightly different principle is used when typesetting radicals.[3] In this case, TeX draws a rule of thickness θ and applies an outside clearance which is also θ, but uses a different value for the inside clearance between $\theta + \frac{1}{4}\sigma_5$ in the bigger sizes and $\frac{5}{4}\theta$ in the smaller sizes, which is not too far away from 3θ and θ either.

As a conclusion, we can assume these values for the amount of clearance as a design principle and try to apply them to other situations where we also have to determine the values of font metric parameters describing some kind of clearance.[4]

4 Typesetting big operators

Let's consider the application of these principles to font metric parameters affecting the placement of limits above or below big operators (see Fig. 1).

Altogether, there are five parameters, ξ_9 to ξ_{13}, whose values we have to determine. In particular, ξ_9 and ξ_{10} affect the inside clearance between the operator and the upper and lower limits, while ξ_{13} affects the outside clearance on both sides.

In the example of Computer Modern fonts, the following values are used: $\xi_9 = {}^{40}\!/_{36}\,\mathrm{pt} \approx 1.11\,\mathrm{pt}$, $\xi_{10} = {}^{60}\!/_{36}\,\mathrm{pt} \approx 1.66\,\mathrm{pt}$, $\xi_{13} = 1.0\,\mathrm{pt}$.

Setting all of them to $3\theta = 1.2\,\mathrm{pt}$ would only make a minor difference, so this approach could be good enough as a starting point for new fonts, where we don't have to worry about compatibility.

[3] To be precise, it should be noted that the thickness of the rule above of the radicals does not directly depend on the rule thickness, but is actually determined by the height of the radical glyph, which is typically designed to be exactly the height of the default rule thickness $\theta = \xi_8$.

[4] Incidently, in the file `tex82.bug`, Don Knuth suggests that there should be additional font metric parameters to govern the space between the rule and the text in cases such as overlines or radicals. Font designers having to define the values of all such parameters might disagree.

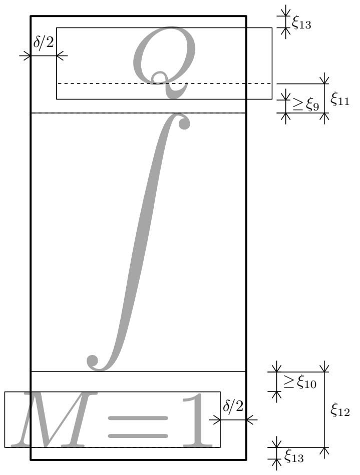

Figure 1: Font metric parameters affecting the placement of limits above or below big operators.

In addition, it would also be more consistent to use the same values for ξ_9 and ξ_{10}, instead of having a different amount of clearance above and below the operator for the upper and lower limit.

As for the outside clearance, represented by ξ_{13}, setting it to 3θ would be closer to the current value, but setting it to θ would be more consistent with the approach used in overlines and radicals.

Once we have determined ξ_9 and ξ_{10} for the boundary case, we can easily determine the values of ξ_{11} and ξ_{12} for the standard case.

Taking into account the way these parameters are measured (see Fig. 1), we just have to take the minimum values given by ξ_9 and ξ_{10} and add the descender depth or ascender height for the typical font size used to typeset the upper and lower limits of big operators:[5]

$$\xi_{11} = \xi_9 \;\; + \tfrac{7}{10} \cdot \mathit{desc\_depth}$$
$$\xi_{12} = \xi_{10} + \tfrac{7}{10} \cdot \mathit{asc\_height}$$

Assuming $\xi_9 = \xi_{10} = 3\theta$ and inserting the values of $\mathit{desc\_depth} = {}^{70}\!/_{36}\,\mathrm{pt}$ and $\mathit{asc\_height} = {}^{250}\!/_{36}\,\mathrm{pt}$, we arrive at the values $\xi_{11} \approx 2.56\,\mathrm{pt}$, $\xi_{12} \approx 6.06\,\mathrm{pt}$, which have to be compared to the reference values $\xi_{11} = 2.0\,\mathrm{pt}$, $\xi_{12} = 6.0\,\mathrm{pt}$ in Computer Modern.

Again, the values determined by our approach would only make a minor difference and could be good enough as a starting point for new fonts.

[5] The factor $\frac{7}{10}$ is based on the assumption that a 7 pt font size is used to typeset the upper and lower limits of big operators at a 10 pt design size. Different factors have to be applied for different design sizes, e. g. $\frac{5}{7}$ for a 7 pt design size. Moreover, the factor is also based on the assumption that *asc_height* and *desc_depth* scale linearly with the design size.

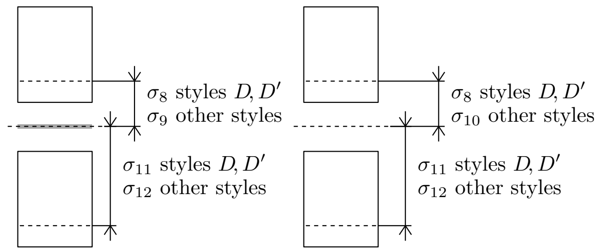

Figure 2: Font metric parameters affecting the placement of numerators and denominators in regular and generalized fractions with and without a fraction bar.

5 Typesetting generalized fractions

Let's now consider the font metric parameters affecting the placement of numerators and denominators in generalized fractions (see Fig. 2).

Apart from the boundary cases, where the minimum shift amounts are determined by built-in rules, there are five parameters, σ_8 to σ_{12}, whose values we have to define: σ_8 and σ_{11} apply to the numerator and denominator in the bigger sizes (display style), σ_9 and σ_{12} apply to the numerator and denominator in the smaller sizes (text style and below). Finally, σ_{10} applies to the numerator in the case of a generalized fraction when the fraction bar is absent, which may or may not require extra adjustments.

Taking into account the way these parameters are measured (see Fig. 2) and considering the total clearance required for the boundary case, we have a total clearance of 7θ that has to be distributed between σ_8 and σ_{11} and a total clearance of 3θ that has be distributed between σ_9 and σ_{12}.

If we assume the math axis as the reference point, the resulting total clearance would be distributed evenly on both sides, resulting in an offset of $\pm 3.5\theta$ or $\pm 1.5\theta$ relative to the math axis height. In addition, we again have to add the descender depth or ascender height for the typical font size used in the numerator or denominator.

For σ_{10} one might want to add an extra amount of clearance of θ to compensate for the absence of a fraction bar of thickness θ, but strictly speaking this shouldn't be necessary, so we could just as well set $\sigma_{10} = \sigma_9$ directly.

In total, we thus arrive at the following values as a first approximation for the font metric parameters:

$$\sigma_8 \;\; = \mathit{math\_axis} + 3.5\theta + \tfrac{7}{10} \cdot \mathit{desc\_depth}$$
$$\sigma_9 \;\; = \mathit{math\_axis} + 1.5\theta + \tfrac{7}{10} \cdot \mathit{desc\_depth}$$

and

$$\sigma_{11} = -\bigl(\mathit{math\_axis} - 3.5\theta - \tfrac{7}{10} \cdot \mathit{asc\_height}\bigr)$$
$$\sigma_{12} = -\bigl(\mathit{math\_axis} - 1.5\theta - \tfrac{7}{10} \cdot \mathit{asc\_height}\bigr)$$

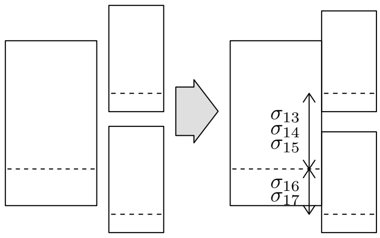

Figure 3: Font metric parameters affecting the placement of superscripts and subscripts on a character box.

Comparing these results to the values used in the sources of the Computer Modern fonts, we find that a similar construction is indeed used, but differs in some details. While some of the terms are omitted, some additional terms are also added.

In the case σ_9 and σ_{10}, the offset to compensate for the descender depth of the numerator is omitted in Computer Modern, possibly based on the assumption that the numerator of fractions in small sizes is just a simple expression without a descender, while the numerator of fractions in bigger sizes is more likely to contain complicated expressions.

It remains to be seen from test cases, if adding or omitting such an offset will produce better results. If the numerator is a simple expression without a descender (such as in $\frac{7}{10}$), including the offset might well produce too much whitespace, whereas omitting it should not make much of a difference.[6]

In addition to the minimum values calculated above, the values used in Computer Modern fonts also add some extra offsets, which has the effect of opening up typeset fractions a little further than the minimum required in the boundary case.

These extra offsets may appear to be arbitrary, but they have an interesting property: Unlike the multiples of rule thickness which have a fixed size, these extra offsets can vary in different design sizes and thus might represent a second order correction in the process of fine-tuning the output quality.

Perhaps the best approach for setting up new fonts could be to start with the above-mentioned values as a first approximation and try them on some documents to see how they work out. If the results aren't good enough, it is always possible to apply additional corrections to improve the spacing.

6 Typesetting superscripts and subscripts

Let's move on to the parameters affecting the placement of superscripts and subscripts (see Fig. 3).

Unlike the previous sections, which could be discussed in terms of establishing the amount of clearance, the parameters in this section have to be discussed in terms of alignment.

On the one hand, we have some constraints for the maximum or minimum shift amounts relative to the x-height, which are intended to ensure that the superscripts or subscripts are properly attached, but also clearly recognizable as being raised or lowered relative to the base glyph.[7]

On the other hand, we also have constraints for the maximum shift amount for raising a superscript or lowering a subscript, so that the resulting expression does not interfere with the interline spacing of body text when it appears in inline math.

Apart from the boundary cases determined by built-in rules, there are five parameters, σ_{13} to σ_{17}, which apply when superscripts and subscripts are attached to a character box.

For the superscripts there are several choices depending on size: σ_{13} applies in the bigger sizes (display style), σ_{14} applies in the smaller sizes (text style and below), and σ_{15} applies in the so-called cramped styles, e. g. when superscripts appear under a fraction bar or a radical.

For the subscripts there are different choices: σ_{16} applies when subscripts appear by themselves, while σ_{17} applies when subscripts appear together with superscripts.

The idea behind this reasoning is that superscripts can be raised a little higher when there is more space available, while they can't be raised as much when they appear in inline math and have to respect a maximum height, so as not to interfere with interline spacing.

Similarly, the subscripts do not have to be lowered as much when they appear by themselves, while they have to be lowered a little further when a superscript is also present, so as to ensure a minimum clearance between superscripts and subscripts.

How can go about to determine the values of font metric parameters under these circumstances? Probably the best approach is to study the constructions used in the sources of the Computer Modern fonts in order to understand the idea behind it.

[6] If it turns out that the offset is indeed better omitted, a similar correction might also have to be applied to the spacing of big operators, setting $\xi_{11} = \xi_9$.

[7] Specifically, these constraints imply that the bottom of the superscript cannot be placed lower than $\frac{1}{4}\sigma_5$ and that the top of the subscript cannot be placed higher than $\frac{4}{5}\sigma_5$ when they appear by themselves. In addition, the bottom of the superscript cannot be placed higher than $\frac{4}{5}\sigma_5$ when both a superscript and script are present with a minimum required clearance of 4θ in between. See [3, 4] for detailed illustrations of these constraints. (Note that σ_5 represents the x-height, which is the minimum size of a base glyph such as x in expressions like x_0 or x_2.)

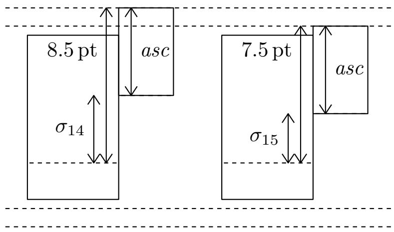

Figure 4: Construction to determine the placement of superscripts relative to a certain maximum height in the case of inline math (σ_{14}) or cramped style (σ_{15}). The dashed lines indicate the height and depth of tallest glyphs and the available interline spacing.

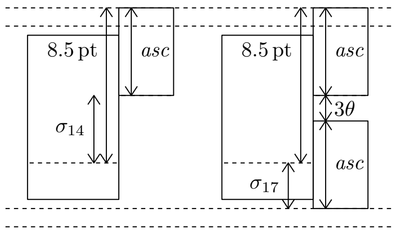

Figure 5: Construction to determine the placement of subscripts relative to the position of superscripts and the minimum required clearance in between. For simplicity, it is assumed that the superscripts and subscripts are simple expressions without descenders.

As it turns out, the values of σ_{13} and σ_{14} are not actually constructed in terms of measuring upwards from the baseline, but downwards from a certain maximum height, which is fixed at 9.0 pt for display math or at 8.5 pt for inline math (see Fig. 4).

Considering that the tallest glyphs in a 10 pt font typically extend 7.5 pt above and 2.5 pt below the baseline (centered on the math axis at 2.5 pt), the value of the maximum height of 8.5 pt is carefully chosen, so that the height of the superscripts will exceed the height of tallest glyphs by no more than 1 pt, leaving at least 1 pt of remaining interline space, assuming a typical baseline skip of 12 pt.

Taking into account the way these parameters are measured (see Fig. 3), we start from the desired maximum height (depending on the size) and subtract the ascender height of the typical font size used in superscripts to determine the shift amount of the baseline of the superscripts:

$$\sigma_{13} = 9.0\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc\_height}$$
$$\sigma_{14} = 8.5\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc\_height}$$
$$\sigma_{15} = 7.5\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc\_height}$$

These values apply for the placement of superscripts in display math (σ_{13}), inline math (σ_{14}), or cramped style (σ_{15}) for a 10 pt design size.[8]

For cramped style, a different construction is used in Computer Modern fonts, fixing the baseline of the superscripts at a value of about $\frac{2}{3}$ x-height, instead of calculating it from a maximum height. Nevertheless, we have chosen to apply the same type of calculation for σ_{15} here as well, hoping to achieve more consistency this way.

[8] Once again, the factor $\frac{7}{10}$ is based on the assumption that a 7 pt font size is used for superscripts of a 10 pt design size. In addition, the maximum height of 8.5 pt in inline math or 7.5 pt in cramped style are based on the assumption of a 12 pt baseline skip and a 1 pt lineskip limit. Once again, different factors and different values of the maximum height have to be applied for different design sizes.

Turning to the parameters affecting the subscripts, the situation is different again. Analyzing the values used in the Computer Modern fonts, we find that for most design sizes arbitrary values are used, which bear little relation to design parameters such as the x-height or descender depth.

A systematic construction is only used when both superscripts and subscripts appear together. In this case, the position of the subscripts can be calculated from the desired position of the superscripts (as determined above) and the required minimum clearance in between (see Fig. 5).[9]

Following this construction, we start from the desired maximum height (applicable for inline math) and subtract twice the ascender height of the font size used in superscripts and subscripts as well as the required clearance to determine the shift amount of the baseline of the subscripts:

$$\sigma_{17} = -\bigl(8.5\,\mathrm{pt} - 2 \cdot \tfrac{7}{10} \cdot \mathit{asc\_height} - 3\theta\bigr)$$

Instead of picking an arbitrary value for σ_{16} when subscripts appear by themselves, we have chosen to set $\sigma_{16} = \sigma_{17}$, using the same value as for subscripts appearing together with superscripts.

This way, we can easily avoid inconsistencies of alignment of subscripts with and without superscripts that would normally appear if different values of σ_{16} and σ_{17} were used.

As it turns out, inserting $\mathit{asc\_height} = {}^{250}\!/_{36}\,\mathrm{pt}$ for Computer Modern fonts, we arrive at a value of $\sigma_{16} = \sigma_{17} \approx 2.42\,\mathrm{pt}$, which almost happens to coincide with the depth of the tallest glyphs, which is typically 2.5 pt below the baseline.

[9] Incidently, this construction exhibits an inconsistency of the built-in rules of TeX, as the minimum clearance between superscript and subscript is required to be at least 4θ while the minimum clearance between numerator and denominator in a generalized fraction without a fraction bar is required to be only 3θ in inline math. In the Computer Modern fonts, the minimum clearance is also assumed to be 3θ.

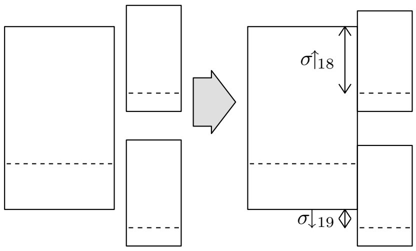

Figure 6: Font metric parameters affecting the placement of superscripts and subscripts on a boxed formula.

Instead of fixing the desired maximum height and calculating the position of subscripts from the position of superscripts and the required minimum clearance in between, we could just as well determine the subscript position directly by fixing the desired maximum depth at 2.5 pt.

Except for setting $\sigma_{16} = \sigma_{17}$, which is a design decision by itself, either method of calculating or fixing the values would make only a minor difference and could be good enough for new fonts.

7 Typesetting superscripts and subscripts of boxed formulas

In addition to the parameters discussed so far, there is another set of two parameters, σ_{18} and σ_{19}, which apply when superscripts or subscripts are attached to a boxed formula (see Fig. 6).

Unlike the other parameters, which are measured as a distance to the baseline, these parameters are measured as an offset to the height and the depth of the boxed formula, which makes them independent of the size of the box.

Once again, consulting the construction used in the sources of the Computer Modern fonts helps to understand the idea behind it.

As it turns out, the idea is to have the height of the superscripts exceed the height of the box by 1 pt and to have the subscripts extend below the depth of the box by 0.5 pt, so that both superscripts and subscripts will always stick out enough to be visible, whatever the size of the box might be.

For the placement of superscripts of a boxed formula, we again have to apply an offset for the ascender height of the superscripts to determine the shift amount relative to the height of the box:

$$\sigma_{18} = -\bigl(1.0\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc\_height}\bigr)$$

For the placement of subscripts, however, we do not have to apply any offset for the descender depth of the subscripts and we have a fixed shift amount of $\sigma_{19} = 0.5\,\mathrm{pt}$ relative to the depth of the box, which is independent of the design size of the font.

8 Typesetting delimiters around fractions

In the previous sections, we have discussed several groups of parameters, covering nearly all of the font metric parameters presented in Table 1.

The only parameters left over are σ_{20} and σ_{21}, affecting the size of delimiters enclosing fractions: σ_{20} applies to the bigger sizes (display math), while σ_{21} applies to the smaller sizes (inline math).

In the Computer Modern fonts, the values of these parameters depend on the font size, but are otherwise fixed based on the available design sizes. Given that the sizes of big delimiters are designed in steps of 10 pt, 12 pt, 18 pt, 24 pt, etc. the values of σ_{20} and σ_{21} are chosen from these sizes.

For display math, the usual choice is to set σ_{20} equal to the one of the bigger sizes, such as using 24 pt delimiters for a 10 pt font size.

For inline math, the usual choice is to set σ_{21} equal to the design size, which means that simple fractions should normally be designed to fit within the default size of 10 pt delimiters.

9 Summary and Conclusions

In the previous sections, we have presented a systematic approach to determine the values of most font metric parameters in terms of very basic design parameters such as the x-height (σ_5), the math axis (σ_{22}), or the default rule thickness (ξ_8).

Except for a few specific design decisions, such as setting $\sigma_{16} = \sigma_{17}$ (to improve the alignment of subscripts), our approach is mostly based on analyzing and understanding the constructions used in existing fonts or drawing conclusions from built-in rules applicable for boundary cases.

In most cases, we have closely tried to follow established practices, except for a few cases where the existing values seemed arbitrary or inconsistent. For this reason, most of the results of our approach will exhibit only minor differences compared to the parameters of existing fonts, except for the few cases where we have tried to improve things by avoiding arbitrary choices or inconsistencies.

We hope that our results will be helpful to font designers facing the task of defining the font metric parameters for new families of math fonts.

In the end, it will always be necessary to test new fonts and it may also be necessary to apply some additional corrections to improve the quality, but having some guidelines of how to set up the font metric parameters might help to get started.

In any event, it is important to make careful choices of the values of the scaling factors or the desired maximum heights, rather than just copying the formulas as they appear in this paper.

Our approach should be understood as a guideline of how to set up new math fonts, but it should be applied thoughtfully and not blindly.

While a typical TEX font family such as Latin Modern (inspired from Computer Modern) will have some meta-ness and multiple design sizes, a typical PostScript font such as Termes or Pagella (inspired from Times or Palatino) will usually have only one design size and will require using scaled-down versions for superscripts and subscripts.

Most likely, this will result in using a different set of sizes (such as 6, 7.6, 10 instead of 5, 7, 10), which will also require using different scaling factors (such as $\frac{7.6}{10}$ or $\frac{6}{7.6}$ instead of $\frac{7}{10}$ or $\frac{5}{7}$).

In addition, different font designs will also have different proportions of x-height, ascender height, descender depth, or math axis position, which could also have side-effects on the calculations.

In summary, it is important to make reasonable choices of scaling factors or offsets as appropriate for the font under development.

Acknowledgments

The author wishes to thank Bogusław Jackowski for permission to reproduce and adapt the figures from his paper *Appendix G Illuminated* [3, 4], which was the main inspiration for writing this paper.

References

[1] Donald E. Knuth: *The TEXbook*. Volume A of *Computers & Typesetting*, 1986.

[2] Donald E. Knuth: *Computer Modern Typefaces*. Volume E of *Computers & Typesetting*, 1986.

[3] Bogusław Jackowski: Appendix G Illuminated. Proceedings of the 16th EuroTEX Conference, Debrecen, Hungary, 77–83, 2006. `http://www.gust.org.pl/projects/e-foundry/math-support/`

[4] Bogusław Jackowski: Appendix G Illuminated. *TUGboat*, 27(1):83–90, 2006. `http://www.tug.org/TUGboat/Articles/tb27-1/tb86jackowski.pdf`

[5] Ulrik Vieth: Math Typesetting in TEX: The Good, the Bad, the Ugly. Proceedings of the 12th EuroTEX Conference, Kerkrade, Netherlands. *MAPS*, 26:207–216, 2001. `http://www.ntg.nl/maps/26/27.pdf`

[6] Ulrik Vieth: Math Typesetting in TEX: The Good, the Bad, the Ugly. *Biuletyn GUST*, 17:24–30, 2001. `ftp://ftp.gust.org.pl/pub/GUST/bulletin/17/vieth01.ps.gz`

A Summary of formulas

In this appendix, we once again summarize all the formulas for obtaining the font metrics parameters of math fonts.

Typesetting big operators

Font metric parameters affecting the placement of limits on big operators can be expressed in terms of the default rule thickness ($\theta = \xi_8$) as follows:

$$\begin{aligned}
\xi_9 &= 3\theta \\
\xi_{10} &= 3\theta \\
\xi_{11} &= 3\theta + \tfrac{7}{10} \cdot \mathit{desc_depth} \\
\xi_{12} &= 3\theta + \tfrac{7}{10} \cdot \mathit{asc_height} \\
\xi_{13} &= 3\theta
\end{aligned}$$

Alternatively, one might set $\xi_{11} = \xi_9$ to achieve more consistent distribution of whitespace if the upper limit appears without descenders.

Typesetting generalized fractions

Font metric parameters affecting the placement of numerators and denominators of fractions can be expressed in terms of the math axis (σ_{22}) and the default rule thickness ($\theta = \xi_8$) as follows:

$$\begin{aligned}
\sigma_8 &= \sigma_{22} + 3.5\theta + \tfrac{7}{10} \cdot \mathit{desc_depth} \\
\sigma_9 &= \sigma_{22} + 1.5\theta + \tfrac{7}{10} \cdot \mathit{desc_depth} \\
\sigma_{10} &= \sigma_{22} + 1.5\theta + \tfrac{7}{10} \cdot \mathit{desc_depth} \\
\sigma_{11} &= -\bigl(\sigma_{22} - 3.5\theta - \tfrac{7}{10} \cdot \mathit{asc_height}\bigr) \\
\sigma_{12} &= -\bigl(\sigma_{22} - 1.5\theta - \tfrac{7}{10} \cdot \mathit{asc_height}\bigr)
\end{aligned}$$

Alternatively, one might set $\sigma_9 = \sigma_{10} = \sigma_{22} + 1.5\theta$ to achieve more consistent distribution of whitespace if the numerator appears without descenders.

In addition, some extra space could be added to $\sigma_9 = \sigma_{10}$ and σ_{12} as a second order correction.

Typesetting superscripts or subscripts

Font metric parameters affecting the placement of superscripts or subscripts can be as follows:

$$\begin{aligned}
\sigma_{13} &= 9.0\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc_height} \\
\sigma_{14} &= 8.5\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc_height} \\
\sigma_{15} &= 7.5\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc_height} \\
\sigma_{16} &= -\bigl(8.5\,\mathrm{pt} - 2 \cdot \tfrac{7}{10} \cdot \mathit{asc_height} - 3\theta\bigr) \\
\sigma_{17} &= -\bigl(8.5\,\mathrm{pt} - 2 \cdot \tfrac{7}{10} \cdot \mathit{asc_height} - 3\theta\bigr) \\
\sigma_{18} &= -\bigl(1.0\,\mathrm{pt} - \tfrac{7}{10} \cdot \mathit{asc_height}\bigr) \\
\sigma_{19} &= 0.5\,\mathrm{pt}
\end{aligned}$$

Alternatively, one might set $\sigma_{16} = \sigma_{17} = 2.5\,\mathrm{pt}$ to match the depth of descenders of the tallest glyphs (usually the delimiters) in a 10 pt font size.

B Examples

In this appendix, some examples are presented to illustrate the effects of manipulating the font metric parameters. Readers are encouraged to consult the online version of this paper at very high magnifications to study the details.

Typesetting big operators

The following examples illustrate the effect of setting the values of ξ_8 to ξ_{13} as discussed in section 4 compared to the original values.

(a) Original values as found in Computer Modern: $\xi_9 \approx 1.11\,\mathrm{pt}$, $\xi_{10} \approx 1.66\,\mathrm{pt}$, $\xi_{11} = 2.0\,\mathrm{pt}$, $\xi_{12} = 6.0\,\mathrm{pt}$, $\xi_{13} = 1.0\,\mathrm{pt}$:

$$\boxed{\sum_{x=1}^{N}} \quad \boxed{\sum_{x=1}^{\xi}}$$

(b) New values calculated as derived in section 4: $\xi_9 = \xi_{10} = \xi_{13} = 1.2\,\mathrm{pt}$, $\xi_{11} \approx 2.56\,\mathrm{pt}$, $\xi_{12} \approx 6.06\,\mathrm{pt}$:

$$\boxed{\sum_{x=1}^{N}} \quad \boxed{\sum_{x=1}^{\xi}}$$

Too much whitespace on the upper limit without descenders.

(c) Same as above, except for $\xi_{11} = \xi_9 = 1.2\,\mathrm{pt}$:

$$\boxed{\sum_{x=1}^{N}} \quad \boxed{\sum_{x=1}^{\xi}}$$

More consistent distribution of whitespace on the upper and lower limit.

Typesetting generalized fractions

The following examples illustrate the effect of setting the values of σ_8 to σ_{12} as discussed in section 5 compared to the original values.

(a) Original values as found in Computer Modern: $\sigma_9 \approx 3.93\,\mathrm{pt}$, $\sigma_{10} \approx 4.43\,\mathrm{pt}$, $\sigma_{12} \approx 3.44\,\mathrm{pt}$:

$$\tfrac{1}{2} \quad \tfrac{1}{2} \quad \tfrac{\xi}{2} \quad \tfrac{\xi}{2} \quad \boxed{\tfrac{1}{2}} \quad \boxed{\tfrac{1}{2}} \quad \boxed{\tfrac{\xi}{2}} \quad \boxed{\tfrac{\xi}{2}}$$

(b) New values calculated as derived in section 5: $\sigma_9 = \sigma_{10} = 4.46\,\mathrm{pt}$, $\sigma_{12} \approx 2.96\,\mathrm{pt}$:

$$\tfrac{1}{2} \quad \tfrac{1}{2} \quad \tfrac{\xi}{2} \quad \tfrac{\xi}{2} \quad \boxed{\tfrac{1}{2}} \quad \boxed{\tfrac{1}{2}} \quad \boxed{\tfrac{\xi}{2}} \quad \boxed{\tfrac{\xi}{2}}$$

Too much whitespace on the numerator without descenders.

(c) Same as above, except for $\sigma_9 = \sigma_{10} = 3.1\,\mathrm{pt}$:

$$\tfrac{1}{2} \quad \tfrac{1}{2} \quad \tfrac{\xi}{2} \quad \tfrac{\xi}{2} \quad \boxed{\tfrac{1}{2}} \quad \boxed{\tfrac{1}{2}} \quad \boxed{\tfrac{\xi}{2}} \quad \boxed{\tfrac{\xi}{2}}$$

More consistent distribution of whitespace on the numerator and denominator, but very close setting.

Typesetting superscripts

The following examples illustrate the effect of setting the values of σ_{13} to σ_{15} as discussed in section 6 compared to the original values.

(a) Original values as found in Computer Modern: $\sigma_{13} \approx 4.12\,\mathrm{pt}$, $\sigma_{14} \approx 3.62\,\mathrm{pt}$, $\sigma_{15} \approx 2.88\,\mathrm{pt}$:

$$(x^2) \quad (x^2) \quad \sqrt{(x^2)} \quad \boxed{(x^2)} \quad \boxed{(x^2)} \quad \boxed{\sqrt{(x^{2k})}}$$

(b) New values calculated as derived in section 6: $\sigma_{13} \approx 4.12\,\mathrm{pt}$, $\sigma_{14} \approx 3.62\,\mathrm{pt}$, $\sigma_{15} \approx 2.62\,\mathrm{pt}$:

$$(x^2) \quad (x^2) \quad \sqrt{(x^2)} \quad \boxed{(x^2)} \quad \boxed{(x^2)} \quad \boxed{\sqrt{(x^{2k})}}$$

Only very small differences in cramped style.

Typesetting subscripts

The following examples illustrate the effect of setting the values of σ_{16} to σ_{17} as discussed in section 6 compared to the original values.

(a) Original values as found in Computer Modern: $\sigma_{16} \approx 1.5\,\mathrm{pt}$, $\sigma_{17} \approx 2.47\,\mathrm{pt}$:

$$x_0 \quad x_0 \quad x_0^{2k} \quad \boxed{x_0} \quad \boxed{x_0} \quad \boxed{x_0^{2k}}$$

Note that subscripts without superscripts produce inconsistent alignment compared to subscripts with an empty superscript.

(b) New values calculated as derived in section 6: $\sigma_{16} = \sigma_{17} \approx 2.42\,\mathrm{pt}$:

$$x_0 \quad x_0 \quad x_0^{2k} \quad \boxed{x_0} \quad \boxed{x_0} \quad \boxed{x_0^{2k}}$$

Note that subscripts without a superscript are now lowered by the same amount, so that inconsistent alignment is avoided.