Understanding the æsthetics of math typesetting

Dr. Ulrik Vieth

Stuttgart, Germany

BachoTEX 2008

Quality of math typesetting

- TEX is well-known for its æsthetic quality (in general)
- TEX is well-known for math typesetting (in particular)
- Quality of math typesetting is almost taken for granted
- Quality of math typesetting is not so well-understood
- Quality depends on multiple factors:
  - Algorithms for math typesetting (built into TEX)
  - Font metric parameters (built into math fonts)
  - Interaction between algorithms and font metrics

Algorithms & Font metrics

- Algorithms for math typesetting
  - determine how to do math typesetting
  - built into TEX engines
  - provided by TEX implementors
  - fixed reference implementation (DEK)
  - defined only once and for all
- Font metrics of math fonts
  - determine how much spacing or shifting
  - built into math fonts
  - provided by font designers/implementors
  - example implementations (Computer Modern)
  - needed for each font family, again and again

Documentation & Challenges

- Algorithms are well documented
  - Appendix G of *The TEXbook* by DEK (algorithmic and mathematical description)
  - *Appendix G Illuminated* by BJ, EuroTeX 2006 (visual and geometric description)
- Font metrics are not so well understood
  - meaning of parameters is relative clear (in principle)
  - values of parameters remain a mystery (in detail)
  - no documented procedure for setting parameters
- Challenge for font designers:
  - How to set up font metric parameters?
  - Is there a recipe for obtaining good values?

Understanding font metrics

- Contents of font metrics (TFM files)
  - glyph metrics for each glyph
  - ligature and kerning tables
  - table of font metric parameters
- Requirements for font metric parameters
  - typical text font needs 7 parameters
    (slant, x-height, quad width, interword space)
  - math symbol font needs 22 parameters
    (fractions, subscripts, superscripts, math axis)
  - math extension font needs 13 parameters
    (big operators, rule thickness)

Font metrics of math fonts

- some parameters are determined by font design
  - nominal design size, e.g. 10 pt
  - body height and depth, e.g. $7.5 + 2.5$ pt
  - ascenders and descenders, e.g. $6.94 + 1.94$ pt
  - x-height of lowercase (σ_5), e.g. 4.31 pt
  - math axis position (σ_{22}), e.g. 2.5 pt
  - default rule thickness (ξ_8), e.g. 0.4 pt
- other parameters remain to be defined
  - $\sigma_8 \ldots \sigma_{12}$ for typesetting fractions
  - $\sigma_{13} \ldots \sigma_{19}$ for typesetting indices
  - $\sigma_{20} \ldots \sigma_{21}$ for typesetting delimiters
  - $\xi_9 \ldots \xi_{13}$ for typesetting big operators

How to determine font metrics?

- Apply scientific methods of theory and experiment
- Analyze available data and facts:
  - Study built-in rules of TEX's algorithms
  - Study built-in rules for boundary cases
  - Study parameters of existing fonts (Computer Modern)
- Formulate a theory:
  - Draw conclusions about possible design principles
  - Formulate a recipe how to calculate parameters
- Conduct experiments:
  - Try to apply conclusions: What are the results?
  - Evaluate the conclusions: Are the results any good?

Understanding design principles

- What are the design principles for spacing of math?
- What is the minimum clearance between elements?
- Analyze built-in rules of TeX's algorithms:
  - Typesetting overlines and underlines
  - Typesetting fractions (boundary case)
  - Typesetting radicals (boundary case)
- Apply conclusions to similar situations:
  - Typesetting big operators
  - Typesetting fractions (general case)

Analyze built-in rules of TEX's algorithms

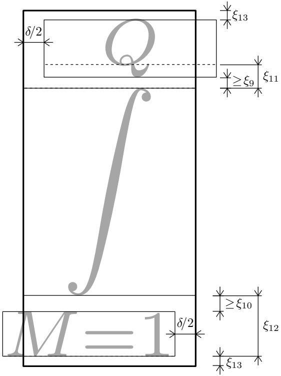
- Typesetting overlines and underlines
  - overline/underline bar: 1 x rule thickness
  - inside clearance: 3 x rule thickness
  - outside clearance: 1 x rule thickness
- Typesetting fractions (boundary case)
  - fraction bar: 1 x rule thickness
  - minimum clearance (display math): 3 x rule thickness
  - minimum clearance (inline math): 1 x rule thickness
- Typesetting radicals (boundary case)
  - radical bar: 1 x rule thickness
  - inside clearance (display math): ≈ 3 x rule thickness
  - inside clearance (inline math): ≈ 1 x rule thickness
  - outside clearance: 1 x rule thickness

Conclusions about design principles

- assume rule thickness as a basic design parameter
- recipe for minimum clearance in boundary cases
  - 1 x rule thickness in inline math
  - 3 x rule thickness in display math
- recipe for default clearance in general case
  - start from minimum values in boundary case
  - adjust for ascender height or descender depth
  - optionally add extra space if result is not optimal

Typesetting big operators (I)

- Parameters for big operator spacing
    - ξ_9 , ξ_{10} minimum inside clearance
    - ξ_{11}, ξ_{12} default inside clearance
    - ξ_{13} outside clearance (both sides)
- Applying design principles
    - $\xi_9 = \xi_{10} = 3\,\mathit{rule\_thickness}$
    - $\xi_{11} = \xi_9 + \frac{7}{10}\,\mathit{desc\_depth}$
    - $\xi_{12} = \xi_{10} + \frac{7}{10}\,\mathit{asc\_height}$
    - $\xi_{13} = 3\,\mathit{rule\_thickness}$

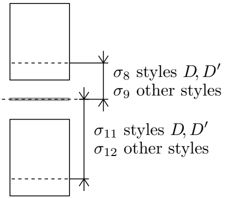


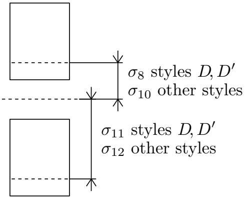
Typesetting big operators (II)

- Comparing results (CM vs. calculated values)
  - $\xi_9 = 1.11$ pt vs. $\xi_9 = 1.20$ pt (minimum clearance)
  - $\xi_{10} = 1.66$ pt vs. $\xi_{10} = 1.20$ pt (minimum clearance)
  - $\xi_{11} = 2.00$ pt vs. $\xi_{11} = 2.56$ pt (default for upper limit)
  - $\xi_{12} = 6.00$ pt vs. $\xi_{12} = 6.06$ pt (default for lower limit)
  - $\xi_{13} = 1.00$ pt vs. $\xi_{13} = 1.20$ pt (outside clearance)
- Evaluating results
  - more systematic approach (symmetric values $\xi_9 = \xi_{10}$)
  - small differences in absolute values
  - no big differences in order of magnitude
  - reasonable starting values for new fonts
  - effect of descenders in upper limit (ξ_{11}) may be too big

Typesetting fractions (I)

- Parameters for fractions
  - σ_8 , σ_9 numerator (display/inline)
  - σ_{11}, σ_{12} denominator (display/inline)
- Boundary cases for total clearance
  - σ_8, σ_{11}: 7 rule thickness (display)
  - σ_9, σ_{12}: 3 rule thickness (inline)
- Applying design principles
  - $\sigma_8 \;=\; (\sigma_{22} + \frac{7}{2}\,\xi_8 + \frac{7}{10}\,\mathit{desc\_depth})$
  - $\sigma_9 \;=\; (\sigma_{22} + \frac{3}{2}\,\xi_8 + \frac{7}{10}\,\mathit{desc\_depth})$
  - $\sigma_{11} = -(\sigma_{22} - \frac{7}{2}\,\xi_8 - \frac{7}{10}\,\mathit{asc\_height})$
  - $\sigma_{12} = -(\sigma_{22} - \frac{3}{2}\,\xi_8 - \frac{7}{10}\,\mathit{asc\_height})$
- Design decision: $\sigma_{10} = \sigma_9$


σ_8 styles D, D'
σ_9 other styles
σ_{11} styles D, D'
σ_{12} other styles


σ_8 styles D, D'
σ_{10} other styles
σ_{11} styles D, D'
σ_{12} other styles

Typesetting fractions (II)

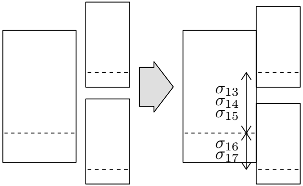
- Comparing results (CM vs. calculated values)
  - $\sigma_8 = 6.76$ pt vs. $\sigma_8 = 5.26$ pt (numerator display math)
  - $\sigma_9 = 3.93$ pt vs. $\sigma_9 = 4.46$ pt (numerator inline math)
  - $\sigma_{10} = 4.43$ pt vs. $\sigma_{10} = 4.46$ pt (numerator w/o fraction bar)
  - $\sigma_{11} = 6.86$ pt vs. $\sigma_{11} = 3.76$ pt (denominator display math)
  - $\sigma_{12} = 3.44$ pt vs. $\sigma_{12} = 2.96$ pt (denominator inline math)
- Evaluating results
  - similar construction used in MF sources of CM fonts
  - additional offsets used in CM fonts (2nd order correction)
  - small differences in absolute values (inline math)
  - bigger differences in absolute values (display math)
  - reasonable starting values for new fonts, but very close
  - additional offsets may be also needed for new fonts
  - effect of descenders in nominator (σ_9, σ_{10}) may be too big

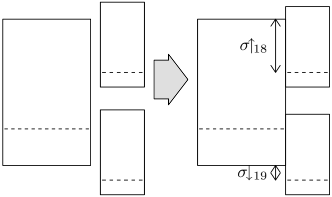
Typesetting superscripts and subscripts (I)

- Different design principles apply in this situation: position and alignment matters more than spacing
  - Minimum shift determined by built-in rules: Superscripts and subscripts should be properly attached
  - Maximum shift constraint by line spacing: Superscripts and subscripts should fit within baseline grid
- Design principles can be deduced from CM fonts
  - Maximum height of superscript is fixed at certain value
  - Position of superscript is calculated relative to maximum
  - Position of subscript is calculated relative to superscript

Typesetting superscripts and subscripts (II)

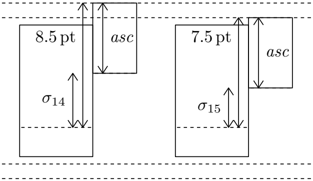
- Parameters for superscripts
  - σ_{13} superscripts (display math)
  - σ_{14} superscripts (inline math)
  - σ_{15} superscripts (cramped style)
- Parameters for subscripts
  - σ_{16} subscripts (w/o superscripts)
  - σ_{17} subscripts (with superscripts)
- Parameters for boxed formulas
  - σ_{18} superscripts on boxes
  - σ_{19} subscripts on boxes





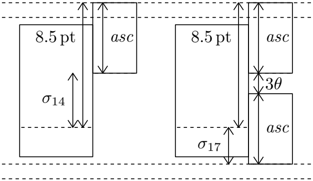
Typesetting superscripts and subscripts (III)

- Recipe for superscript position
  - start from maximum height
  - adjust for height of superscripts
- Applying the recipe
  - $\sigma_{13} = 9.0\,\text{pt} - \frac{7}{10} \cdot \textit{asc\_height}$
  - $\sigma_{14} = 8.5\,\text{pt} - \frac{7}{10} \cdot \textit{asc\_height}$
  - $\sigma_{15} = 7.5\,\text{pt} - \frac{7}{10} \cdot \textit{asc\_height}$
- Understanding the values
  - size of 10 pt delimiters: 7.5 + 2.5 pt constraint for σ_{15} (cramped style)
  - size of 12 pt delimiters: 8.5 + 3.5 pt constraint for σ_{14} (inline math)



Typesetting superscripts and subscripts (IV)

- Recipe for subscript position
  - start from superscript position
  - adjust for minimum clearance
  - adjust for height of subscripts
  - neglect depths of superscripts
- Applying the recipe
  - $\sigma_{17} = \sigma_{14} - 3\xi_8 - \frac{7}{10} \cdot \mathit{asc\_height}$
  - $\sigma_{16} = \sigma_{17}$ (Design decision!)
- Recipe for alternative approach
  - start from minimum depth, e.g. 2.5 pt
  - fix subscript position at minimum depth



Typesetting superscripts and subscripts (V)

- Comparing results (CM vs. calculated values)
  - $\sigma_{13} = 4.12$ pt vs. $\sigma_{13} = 4.12$ pt (superscript display math)
  - $\sigma_{14} = 3.62$ pt vs. $\sigma_{14} = 3.62$ pt (superscript inline math)
  - $\sigma_{15} = 2.88$ pt vs. $\sigma_{15} = 2.62$ pt (superscript cramped style)
  - $\sigma_{16} = 1.50$ pt vs. $\sigma_{16} = 2.42$ pt (subscript w/o superscript)
  - $\sigma_{17} = 2.47$ pt vs. $\sigma_{17} = 2.42$ pt (subscript with superscript)
- Evaluating results
  - similar construction used in MF sources of CM fonts
  - more systematic approach ($\sigma_{16} = \sigma_{17}$)
  - small differences in absolute values
  - no big differences in order of magnitude
  - reasonable starting values for new fonts

Summary and Conclusions

- Challenge for font designers solved (to some degree)
  - Is there a recipe how to set up font metric parameters? Yes!
  - Is there a recipe for obtaining good starting values? Yes!
  - Is there a recipe for obtaining good working values? Maybe?!
  - Do the values obtained need testing? Yes, absolutely!
  - Do the values obtained need adjusting? Yes, probably a bit?!
- Design considerations
  - Basic parameters must always be defined by font designer e.g. x-height (σ_5), math axis (σ_{22}), rule thickness (ξ_8)
  - Repcipe must be applied thoughtfully for each design size e.g. scaling factors $\frac{7}{10}$ for 10 pt, $\frac{5}{7}$ for 7 pt, $\frac{5}{5}$ for 5 pt
  - Recipe must be applied individually for each font family e.g. scaling factors 10/7/5 (CM/LM) vs. 10/7.6/6 (TG)

Discussion

- Discussion points / Design decisions
  - Is there too much space for descenders? Yes, maybe?!
  - Conflicting design goals: Equal spacing or equal alignment?
    - w/o correction for descenders: equal spacing
    - with correction for descenders: equal aligment
  - Should there be extra space on generalized fractions? No?!
    - uneven alignment for $\sigma_{10} > \sigma_9$: Compare $\frac{1}{2}$ vs. $\frac{1}{2}$
    - better alignment for $\sigma_{10} = \sigma_9$: Compare $\frac{1}{2}$ vs. $\frac{1}{2}$
  - Should there be different placement of subscripts? No?!
    - uneven alignment for $\sigma_{16} < \sigma_{17}$: Compare x_0 vs. x_0^2
    - better alignment for $\sigma_{16} = \sigma_{17}$: Compare x_0 vs. x_0^2