Optimizing OpenType math fonts for inline setting

Ulrik Vieth
Stuttgart, Germany

BachoTEX 2016
Brodnica, Poland

About this talk

- Previous studies on math font parameters:
  - Appendix G of *The TEXbook* by Don Knuth (algorithmic description of TEX math font parameters)
  - *Appendix G Illuminated* by B. Jackowski, 2006 (geometric visualization of TEX math font parameters)
  - *Understanding the aesthetics of math* by U. Vieth, 2008 (analytic study of TEX math font parameters)
  - *OpenType Math Illuminated* by U. Vieth, 2009 (visualization of OpenType math font parameters)
- Conclusions:
  - the meaning of math parameters is well understood
  - the choice of parameter values is less well understood

About this talk

- Today's topic:
  - Study of specific aspects of math font parameters
  - What makes a math font work well (or not so well) for inline math typesetting?
- Basic assumptions:
  - typesetting on a baseline grid (e.g. 10 pt on 12 pt)
  - we cannot expect to support arbitrary inline math
  - we want to support only some simple inline math
    - simple superscripts (x^2) or subscripts (x_0)
    - if possible: combined super- and subscripts (x_0^2)
    - if possible: simple inline fractions ($\frac{1}{2}x$)
  - we don't want math to interfere with line spacing

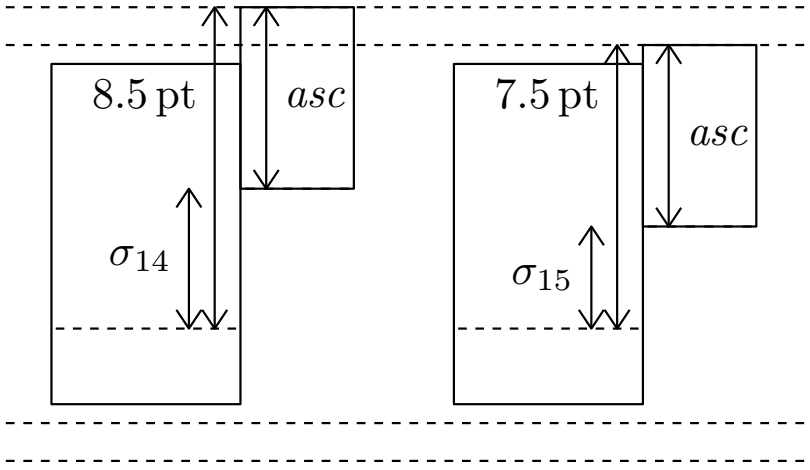
Understanding font sizes

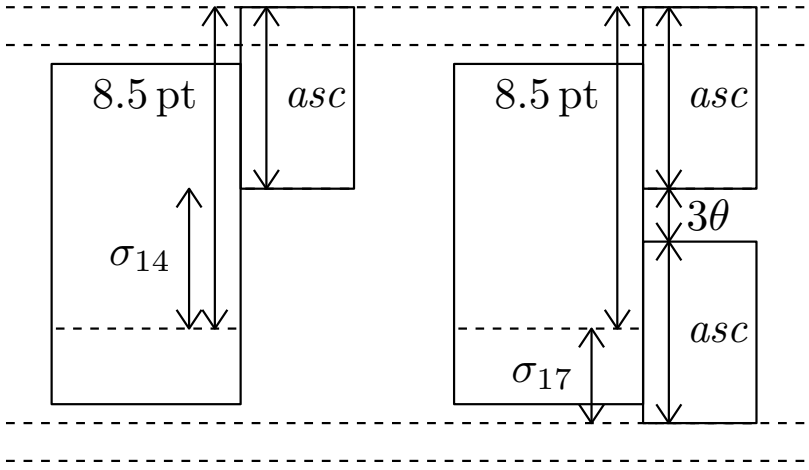
- What does a 10 pt font size mean?
  - only the biggest glyphs (delimiters) are exactly 10 pt
  - most glyphs (letters, numerals, etc) are smaller than 10 pt
- Typical values:
  - delimiters centered on math axis at 2.5 pt height
  - delimiters designed to be 7.5 pt high and 2.5 pt deep
  - most glyphs are less than 7.0 pt high and 2.0 pt deep
  - superscript/subscripts may go 8.5 pt high and 2.5 pt deep

Understanding font sizes

- What does a 10 pt on 12 pt baseline grid mean?
  - nominal font size of 10 pt on 12 pt `\baselineskip`
  - typical setting of `\lineskiplimit` and `\lineskip` to 1 pt
  - no effect if distance between lines is bigger than 1 pt
  - otherwise TEX inserts extra space of 1 pt between lines
- Boundary conditions for inline math:
  - linespacing not affected if math doesn't stick out too much (less than 11 pt total height + depth)
- Alternative approach:
  - ignore `\lineskiplimit` (set it to 0 pt)
  - linespacing not affected if lines do not actually touch (less than 12 pt total height + depth)

How does it work in traditional TEX fonts?





Recipe for superscript position:

- start from maximum height (inline: 8.5 pt, cramped: 7.5 pt)
- adjust for height of superscripts (*asc_height*, scaled to 7 pt)

Recipe for subscript position:

- start from superscript position (as calculated before)
- adjust for minimum clearance (actually 4 not 3 *rule_thickness*)
- adjust for height of subscripts (*asc_height*, scaled to 7 pt)

Why does it work in traditional TEX fonts?

- How much space is available for inline math?

  $$\texttt{\baselineskip} - \texttt{\lineskiplimit} = 11\,\text{pt}$$

- How much space is needed for combined scripts?

  $$4\,\mathit{rule\_thickness} + 2\,\mathit{asc\_height}\ \text{(at 7 pt)}$$

- For LM fonts, it just works by design:

  $$11\,\text{pt} - 4 \cdot 0.4\,\text{pt} = 9.4\,\text{pt} \leq 2 \cdot \frac{7}{10} \cdot 6.44\,\text{pt}$$

- Why is this possible for LM fonts?
  - it only works because we have 7 pt script size
  - it only works because we have optical design sizes otherwise 5 pt script script size might be unreadable

How can it work in various OpenType fonts?

- What happens for other math fonts?
  - most math fonts don't support optical design sizes
  - size of 1st level scripts depends 2nd level scripts
  - size scaling of math fonts often limited by readability

Latin Modern Math:	10 / 7.0 / 5.0 pt
Lucida Math:	10 / 7.0 / 5.0 pt
Cambria Math:	10 / 7.3 / 6.0 pt
TG Termes / Pagella:	10 / 7.4 / 5.5 pt
TG Schola / Bonum:	10 / 7.7 / 6.0 pt
XITS / STIX Math:	10 / 7.5 / 6.0 pt

How can it work in various OpenType fonts?

- What is possible for fonts with bigger sizes?
  - we could adjust position for simple superscripts (x^2)
    (same upper limit at 8.5 pt, but compensate for bigger sizes)
  - we could set fixed position for simple subscripts (x_0)
    (fixed lower limit at 2.5 pt)
- What may be possible for fonts with bigger sizes?
  - combined super-/subscripts (x_0^2) may exceed available size
    (minimum clearance may cause additional shift out)
  - we could try to reduce clearance to stay within bounds
- What may not be possible for fonts with bigger sizes?
  - simple inline fractions ($\frac{1}{2}x$) often exceed available size
  - we could try to reduce clearance, but it may not be enough

What happens in practice (super-/subscripts)?

Math Font	ht x^2	dp x_0	ht x_0^2 + dp x_0^2
LM Math (TFM):	8.14 pt	1.50 pt	8.14 + 2.48 pt
LM Math (OT):	8.28 pt	2.61 pt	8.28 + 2.76 pt
Cambria Math:	8.26 pt	2.11 pt	8.33 + 2.40 pt
TG Termes:	8.40 pt	2.32 pt	8.40 + 3.11 pt
TG Pagella:	8.64 pt	2.46 pt	8.64 + 3.33 pt
TG Bonum:	8.66 pt	2.41 pt	8.66 + 3.48 pt
TG Schola:	8.85 pt	2.51 pt	8.85 + 3.35 pt
XITS Math:	9.05 pt	2.60 pt	9.05 + 3.79 pt
Lucida Math:	9.39 pt	1.62 pt	9.43 + 2.68 pt

What happens in practice (combined scripts)?

Math Font	ht x_0^2 + dp x_0^2
LM Math (TFM):	8.14 + 2.48 = 10.32 pt
LM Math (OT):	8.28 + 2.76 = 11.04 pt
Cambria Math:	8.33 + 2.40 = 10.73 pt
TG Termes:	8.40 + 3.11 = 11.51 pt
TG Pagella:	8.64 + 3.33 = 11.97 pt
TG Bonum:	8.66 + 3.48 = 12.14 pt
TG Schola:	8.85 + 3.35 = 12.20 pt
XITS Math:	9.05 + 3.79 = 12.84 pt
Lucida Math:	9.43 + 2.68 = 12.11 pt

What happens in practice (inline fractions)?

Math Font	ht $\frac{1}{2}$ + dp $\frac{1}{2}$
LM Math (TFM):	8.45 + 3.45 = 11.90 pt
LM Math (OT):	8.60 + 3.45 = 12.05 pt
TG Termes:	9.31 + 3.77 = 13.08 pt
TG Bonum:	9.18 + 4.08 = 13.16 pt
TG Schola:	9.35 + 4.15 = 13.50 pt
TG Pagella:	9.82 + 4.12 = 13.94 pt
XITS Math:	9.85 + 4.80 = 14.65 pt
Lucida Math:	10.11 + 3.60 = 13.71 pt
Cambria Math:	10.43 + 5.03 = 15.46 pt

Observations about space requirements

- scaling of script sizes not very exact in test cases:
  - tested with LuaLaTeX, `unicode-math` and `luatotfload`
  - expected font sizes, e.g. 7.0 pt, 7.3 pt, 7.5 pt, 7.7 pt
  - actual font sizes, e.g. 7.01 pt, 7.27 pt, 7.47 pt, 7.67 pt
- OpenType behavior can differ from TeX behavior:
  - TeX might always insert 4 *rule_thickness* clearance
  - OpenType fonts can set smaller values for clearance
- OpenType math fonts behave very inconsistently:
  - most fonts are not optimized (yet) to stay within bounds
  - some fonts very compact for scripts, but not for fractions
  - different variants of Times are spaced very differently

Related topic: Size scaling of delimiters

- Typical scaling of delimiters (height + depth), centered symmetric on math axis at 2.5 pt

10 pt	(nominal size)	7.5 + 2.5 pt
12 pt	(1.0 baselines)	8.5 + 3.5 pt
18 pt	(1.5 baselines)	11.5 + 6.5 pt
24 pt	(2.0 baselines)	14.5 + 9.5 pt
30 pt	(2.5 baselines)	17.5 + 12.5 pt

Related topic: Size scaling of delimiters

- How TEX determines the size of delimiters:
  - measure required size of expression to be enclosed
  - take maximum of (height - axis) or (axis - depth)
  - find sufficiently big delimiter from available sizes
- How special cases influence the calculation:
  - `\delimitershortfall` (s), e.g. 0.5 pt (absolute offset), i.e. delimiters may be 0.5 pt shorter than required size
  - `\delimiterfactor` (f), e.g. 900/1000 (relative offset) i.e. delimiters need to be only 90% of required size

How to determine sizes (for combined scripts)?

Math Font	axis	ht + dp	$h-a$	$a-d$	required
Cambria Math:	2.85	8.33 + 2.40	5.48	5.25	5.48
LM Math (TFM):	2.50	8.14 + 2.48	5.64	4.98	5.64
LM Math (OT):	2.50	8.28 + 2.76	5.78	5.26	5.78
TG Termes:	2.50	8.40 + 3.11	5.90	5.61	5.90
TG Pagella:	2.50	8.64 + 3.33	6.14	5.73	6.14
TG Bonum:	2.60	8.66 + 3.48	6.16	5.98	6.16
TG Schola:	2.60	8.85 + 3.35	6.25	5.95	6.25
Lucida Math:	3.13	9.43 + 2.68	6.30	5.81	6.30
XITS Math:	2.50	9.05 + 3.79	6.55	6.29	6.55

How to determine sizes (for combined scripts)?

Math Font	required	$2r - s$	$2r\,f$	max	effective
Cambria Math:	5.48	10.46	9.86	10.46	12
LM Math (TFM):	5.64	10.78	10.15	10.78	12
LM Math (OT):	5.78	11.06	10.40	11.06	12
TG Termes:	5.90	11.30	10.62	11.30	12
TG Pagella:	6.14	11.78	11.05	11.78	12
TG Bonum:	6.16	11.82	11.09	11.82	12
TG Schola:	6.25	12.00	11.25	12.00	12
Lucida Math:	6.30	12.10	11.34	12.10	16.5 (> 12)
XITS Math:	6.55	12.60	11.79	12.60	12.8 (> 12)

Observations about delimiter sizes

- Standard settings may not work for all fonts
  - some examples show sudden jumps from 12 pt to 18 pt
  - calculation of delimiter size uses max offset from axis, may need bigger delimiter, even if total size is small enough
  - we may need to increase `\delimitershortfall`, e.g. delimiters may be 1.5 pt short of required size
  - we may need to reduce `\delimiterfactor`, e.g. delimiters need to be only 85% of required size
- How to calculate adjusted settings:
  - delimiter factor = desired size / needed size, e.g. 12.0 pt/13.5 pt = 888
  - elimiter shortfall = needed size - desired size e.g. 13.5 − 12.0 pt = 1.5 pt

Conclusions about space requirements

- To support inline math on a grid:
  - OT fonts should try to stay within bounds (below 12.0 pt)
  - usually no problems with simple super-/subscripts
  - occassional problems with combined scripts (x_0^2)
  - OT fonts could reduce clearance for combined scripts
  - OT fonts could reduce clearance for inline fractions
  - using smaller script sizes may help, if readability permits
  - optical design sizes may help to improve readablitiy

Conclusions about delimiter sizes

- To support inline math on a grid:
  - OT fonts should try to stay symmetric on math axis
  - OT fonts should try to keep offset from axis below 6.0 pt
  - increasing `\delimiterfactor` (above 0.5 pt) may help
  - reducing `\delimitershortfall` (below 90%) may help
  - more intermediate delimiter sizes (e.g. 12 pt, 15 pt, 18 pt) may reduce visual jumps, but won't help inline math