Experiences typesetting OpenType math with LuaLaTeX and XeLaTeX

Dr. Ulrik Vieth
Stuttgart, Germany

4th International ConTeXt Meeting, Brejlow, 2010

Overview of this talk

- Review of OpenType math support in TEX Live 2010
  - Technology review: Unicode and OpenType math
  - Engine support, Macro support, Font support
- Experiences testing OpenType math in TEX Live 2010
  - Engine problems, Macro problems, Font problems
  - Observations and problems during testing
  - Solutions and workarounds for known problems
- Comparing the quality of OpenType math typesetting
  - Choices of test platforms to compare
  - Methods of testing and analyzing test results
  - Examples and discussion of test results
  - Summary and conclusions

Review of OpenType math support

- Review of math technology
  - What is Unicode and Unicode math?
  - What is OpenType and OpenType math?
- Review of math support in TEX Live 2010
  - Engine support for OpenType math
  - Macro support for OpenType math
  - Font support for OpenType math
- Examples of OpenType math usage
  - How does in work?
  - How does it look like?

Technology: Unicode math

- What ist Unicode?
  - encoding standard for (input) characters and symbols
  - maintained by International Unicode Consortium
- What is Unicode math?
  - addition of math symbols and alphabets to Unicode
  - coordinated by STIX group of publishers (late 1990s)
- What does Unicode math provide?
  - hundreds of math symbols added to slots U+2xxx
  - dozens of math alphabets added to slots U+1Dxxx
  - size variants of math symbols are *not* encoded:
    different size, same symbol ⇒ same meaning
  - font variants of math alphabets *are* encoded:
    different font, same letter ⇒ different meaning

Technology: OpenType math

- What ist OpenType?
  - font technology for (output) glyphs and symbols
  - developed jointly by Microsoft and Adobe
  - synthesis of TrueType and PostScript fonts
- What is OpenType math?
  - addition of MATH table to OpenType font format
  - developed by Microsoft for Office 2007 (experimental)
- What does OpenType math provide?
  - global font metrics parameters for spacing of math
  - additional glyph metrics for positioning of math accents
  - lookups for horizontal/vertical variants/constructions
  - base glyphs addressed directly by Unicode slots
  - variant glyphs addressed indirectly through lookups

Engine support for OpenType math

- Microsoft Office 2007, 2010
  - original reference implementation of OpenType math
  - many concepts derived from traditional TEX concepts
  - obvious generalizations and extensions added
- XeTEX
  - support for OpenType math since XeTEX 0.97 (2007)
  - uses mapping of OpenType to TEX parameters
  - makes only limited use of OpenType math features
  - generally available since TEX Live 2008
- LuaTEX
  - support for OpenType math since LuaTEX 0.40 (2009)
  - uses combined set of OpenType and TEX parameters
  - aims to provide full support of OpenType math features
  - generally available since TEX Live 2009

Macro support for OpenType math

- XeLaTeX
  - `fontspec` for high-level text font selection (since long)
  - `unicode-math` for math font selection (since 2008)
- ConTeXt MkIV
  - support for OpenType math since LuaTeX 0.40 (2009)
- LuaTeX (generic)
  - `luaotfload` for low-level font loading (since 2009)
  - no high-level support for OpenType math available
- LuaLaTeX + XeLaTeX
  - `fontspec` for high-level text font selection (revised 2010)
  - `unicode-math` for math font selection (revised 2010)
  - `luaotfload` for low-level font loading (used internally)
  - generally available since TeX Live 2010

Font support for OpenType math

- Cambria Math
  - commissioned by Microsoft, shipped with Office 2007
  - original reference example of OpenType math font
- Asana Math (Apostolos Syropoulos)
  - Palatino-like font derived from `mathpazo` glyphs
  - available from CTAN, included in TeX Live 2010
- XITS Math (Khaled Hosny)
  - Times-like font derived from STIX Fonts 1.0
  - repackaging of raw glyphs with additional metrics
  - available from CTAN, included in TeX Live 2010
- Neo Euler (Khaled Hosny)
  - derived from redesign of AMS Euler (Hermann Zapf)
  - available from GitHub only (still under development)

How does it work?

- ConTEXt MkIV
  - support for OpenType math included in ConTEXt kernel
  - predefined typescripts for Cambria, Asana, XITS, etc.

```
\usetypescript[cambria]
\switchtobodyfont[cambria,10pt]
```

- LuaLaTEX + XeLaTEX
  - support for OpenType math by `unicode-math` package
  - no predefined font sets, fonts loaded on demand
  - configuration options to customize math style, etc.

```
\documentclass{article}
\usepackage{fontspec,unicode-math}
\setromanfont[Ligatures=TeX]{Cambria}
\setmathfont[math-style=ISO]{Cambria Math}
```

How does it look like?

- 3 choices of TEX engines + macro packages
  - LuaTEX with ConTEXt
  - LuaTEX with LuaLaTEX
  - XeTEX with XeLaTEX
- 4 choices of OpenType text + math fonts
  - Cambria + Cambria Math
  - XITS + XITS Math
  - Pagella + Asana Math
  - Pagella + Neo Euler
- 3 x 4 sample pages (one for each combination)

Experiences testing OpenType math

- Experiences testing OpenType math
  - Testing of OpenType math = Testing of a complex system
- Possible causes of testing problems
  - Problems with TEX engines
  - Problems with OpenType fonts
  - Problems with macro packages
  - Font-loading problems
- Examples of testing problems
- Known problems and workarounds

Problems with TEX engines

- fatal engine problems (crashes)
  - XeTEX: 64-bit binaries crashing with segfaults
    unresolved in TEX Live 2010 (too late for pretest)
    workaround: use 32-bit binaries instead of 64-bit
- other engine issues (mis-features)
  - LuaTEX: incorrect size of delimiters for Asana Math
    fixed in LuaTEX 0.61 (but not in TEX Live 2010)
  - XeTEX: inconsistent alignment of super-/subscripts
    unresolved in XeTEX engine, still to be fixed
    workaround: add empty groups to math input

Problems with OpenType fonts

- fatal font problems (broken fonts)
  - broken OpenType fonts can cause engine crashes
    some fixes in LuaTeX 0.61-0.62 (but not in TeX Live 2010)
- other font issues (mis-features)
  - LuaTeX: incorrect size of display operators for Cambria Math
    caused by incorrect font metrics (DisplayOperatorMinHeight)
    already fixed in ConTeXt, still to be fixed in `luaotfload`
  - incorrect shape of default partial sign (upright vs. italic)
    caused by inconsistencies in Unicode font tables
    unlikely to be fixed in OpenType fonts anytime soon

Problems with macro packages

- interaction problems (between engines and macros)
  - XeTeX: incorrect usage of OpenType math font parameters
    `unicode-math` loads OpenType math font into family 4
    XeTeX expects to find math parameters in families 2+3
    XeTeX uses incorrect parameters from CMSY and CMEX

    unresolved in TeX Live 2010, still to be fixed
    workaround: redefine math fonts in families 2+3
- other macro issues (mis-features)
  - `unicode-math`: `\hbar` macro uses CMR font for bar accent
    workaround: use `\hslash` instead of `\hbar` in math
  - ConTeXt: `\hbar` uses diacritic text character (U+0127)
    workaround: use `\hslash` instead of `\hbar` in math

Problems with TEX engines (illustrations)

- LuaTEX: incorrect size of delimiters for Asana Math

$$\gamma^\alpha \left(\frac{\hbar}{\mathrm{i}} \partial_\alpha - q A_\alpha \right) \psi + m_0 c \, \psi = 0 \,,$$

$$\gamma^\alpha \left(\frac{\hbar}{\mathrm{i}} \partial_\alpha - q A_\alpha \right) \psi + m_0 c \, \psi = 0 \,.$$

- XeTEX: inconsistent alignment of super-/subscripts

$$R^{\mu\nu} - \frac{1}{2} R g^{\mu\nu} + \Lambda g^{\mu\nu} = -\frac{8\pi G}{c^2} M^{\mu\nu}$$

$$R^{\mu\nu} - \frac{1}{2} R g^{\mu\nu} + \Lambda g^{\mu\nu} = -\frac{8\pi G}{c^2} M^{\mu\nu}$$

Problems with OpenType fonts (illustrations)

- LuaTeX: incorrect size of display operators for Cambria Math

$$\int_F \varepsilon_0 \boldsymbol{E} \cdot \mathrm{d}\boldsymbol{f} = \int_V \lambda \, \mathrm{d}V, \quad \int_F \boldsymbol{B} \cdot \mathrm{d}\boldsymbol{f} = 0\,,$$

$$\int_F \varepsilon_0 \boldsymbol{E} \cdot \mathrm{d}\boldsymbol{f} = \int_V \lambda \, \mathrm{d}V, \quad \int_F \boldsymbol{B} \cdot \mathrm{d}\boldsymbol{f} = 0\,.$$

- incorrect shape of default partial sign (upright vs. italic)

$\partial \quad \partial \quad \boldsymbol{\partial} \quad \boldsymbol{\partial}$ Cambria Math

$\partial \quad \partial \quad \boldsymbol{\partial} \quad \partial$ XITS Math

$\partial \quad \partial \quad \boldsymbol{\partial} \quad \partial$ Asana Math

Font-loading issues

- Font loading in XeTeX:
  - uses `fontconfig` library to locate OpenType fonts
  - may need to edit `fonts.conf` to add `texmf-local` tree
  - may need to run `fc-cache` to refresh font cache
- Font loading in LuaTeX:
  - uses `kpathsea` library to locate OpenType fonts
  - uses `fonts/opentype` to load `*.otf` fonts (e.g. Euler)
  - uses `fonts/truetype` to load `*.ttc` fonts (e.g. Cambria)
  - uses Lua-based font cache (`luaotfload`)
- Font loading in ConTeXt:
  - uses Lua-based file cache and font cache (`luatools`)
  - does not use `fontconfig` or `kpathsea` libraries
  - uses `fonts.conf` to locate font path of system fonts

Comparing the quality of OpenType math

- Choices of test platforms to compare
- Methods of testing or sampling
- Methods of analyzing test results
- Examples and discussion of test results
- Summary and conclusions

Comparing the quality of OpenType math

- 3 choices of TEX engines + macro packages
  - LuaTEX with ConTEXt
  - LuaTEX with LuaLaTEX
  - XeTEX with XeLaTEX
- 2-3 choices of comparing the quality
  - Comparing LuaLaTEX vs. ConTEXt
  - Comparing LuaLaTEX vs. XeLaTEX

Comparing the quality of OpenType math

- Comparing LuaLaTeX vs. ConTeXt:
  - different user interface (`unicode-math` vs. ConTeXt)
  - similar font loading code (`luaotfload` vs. ConTeXt)
  - same underlying TeX engine (LuaTeX in both cases)
  - same implementation of OpenType math algorithms
- Expectations
  - similar input expected to produce similar output
  - BUT: cannot use identical input for test document
- Usage
  - primarily used for verification of bugs/features

Comparing the quality of OpenType math

- Comparing LuaLaTeX vs. XeLaTeX:
  - same user interface (`unicode-math` in both cases)
  - different font loading code (`luaotfload` vs. XeTeX)
  - different underlying TeX engines (LuaTeX vs. XeTeX)
  - different implementations of OpenType math algorithms
- Expectations
  - same input expected to produce different output
  - BUT: same input can be processed unchanged
- Usage
  - primarily used for discovery and analysis

Methods of testing OpenType math

- Systematic Testing
  - not enough time to do systematic testing of math
  - too much tedious work needed for full test
  - too many possibilities / combinations to test
- Sampling vs. Testing
  - alternative approach: sampling of OpenType math
  - create test document for sampling of notations
  - typeset same test document with each engine
  - typeset same test document with each font
  - inspect and compare test results

Methods of analyzing test results

- Analysis of test results
  - inspect results of test documents for bugs
  - compare results of test documents for quality
- Analyzing large-scale effects
  - large-scale effects easy to find by visual inspection
  - usually caused by problems in engines, fonts, macros
  - usually possible to avoid or fix by workarounds
- Analyzing small-scale effects
  - small-scale effects only visible after fixing large-scale
  - effects can be highlighted in multi-color overlays
  - some effects caused by problems in fonts or macros
  - some effects expected due to engine differences

Examples and discussion of test results (I)

- Test Setup
  - comparing LuaLaTeX and XeLaTeX for several fonts
  - use default setup from TeX Live2010 out-of-the-box
- Observations
  - significant large-scale differences in vertical spacings
  - primarily affects fractions and super-/subscripts
- Explanations
  - caused by incorrect loading of font parameters in XeTeX
  - XeTeXwas using parameters from preloaded CM fonts
  - LuaTeX was using parameters from OpenType math table
- Workarounds
  - correct font loading in macro packages

Examples and discussion of test results (II)

- Test Setup
  - comparing LuaLaTeX and XeLaTeX for several fonts
  - apply workarounds for font loading in XeLaTeX
- Observations
  - fewer large-scale differences in vertical spacing
  - primarily affects spacing of super-/subscripts
- Explanations
  - incorrect alignment of super-/subscripts in XeTeX
  - occurs only for symbols with ascenders/descenders
  - presumably engine bug in XeTeX, unresolved
- Workarounds
  - add empty groups before super-/subscripts

Examples and discussion of test results (III)

- Test Setup
  - comparing LuaLaTeX and XeLaTeX for several fonts
  - apply workarounds for font loading in XeLaTeX
  - apply workarounds for alignment of super-/subscripts
- Observations
  - no more large-scale differences in vertical spacing
  - only small-scale differences in horizontal spacing
- Explanations
  - horizontal spacing affected by italic corrections
  - horizontal spacing affected by math kerning
- Next Steps
  - remaining effects caused by engine differences
  - remaining effects need further detailed analysis

Summary and Conclusions (I)

- Original goal
  - study effects on quality of different implementations
  - XeTeX makes only limited use of OpenType math
  - LuaTeX aims to provide full support of OpenType math
- Observations
  - most large-scale effects caused by bugs (unintentional)
  - remaining small-scale effects smaller than expected
- Next Steps
  - remaining effects obviously caused by engine differences
  - remaining effects need further detailed analysis
  - results of comparison remain inconclusive for now

Summary and Conclusions (II)

- Additional goal
  - find out how OpenType math support in TEX Live works
  - find out if OpenType math support is ready to use
- Observations
  - many problems found during TEX Live 2010 pretest
  - some problems already resolved before release
  - some problems still remain unresolved for now
- Conclusions
  - LuaTEX has fewer known bugs than XeTEX
  - LuaTEX is better supported than XeTEX
  - XeTEX used to have better macro support
  - macro support is equivalent for both engines now

Acknowledgements and Thanks

- Will Robertson (`fontspec`, `unicode-math`)
- Khaled Hosny (`luaotfload`, XITS, NeoEuler)
- Taco Hoekwater (LuaTeX)
- Hans Hagen (ConTeXt)
- Jonathan Kew (XeTeX)
- Karl Berry (TeX Live)
- Peter Breitenlohner (64-bit binaries)