New Font Offerings: Cochineal, Nimbus15, LibertinusT1Math

Michael Sharpe, UCSD

TUG Toronto, July 2016

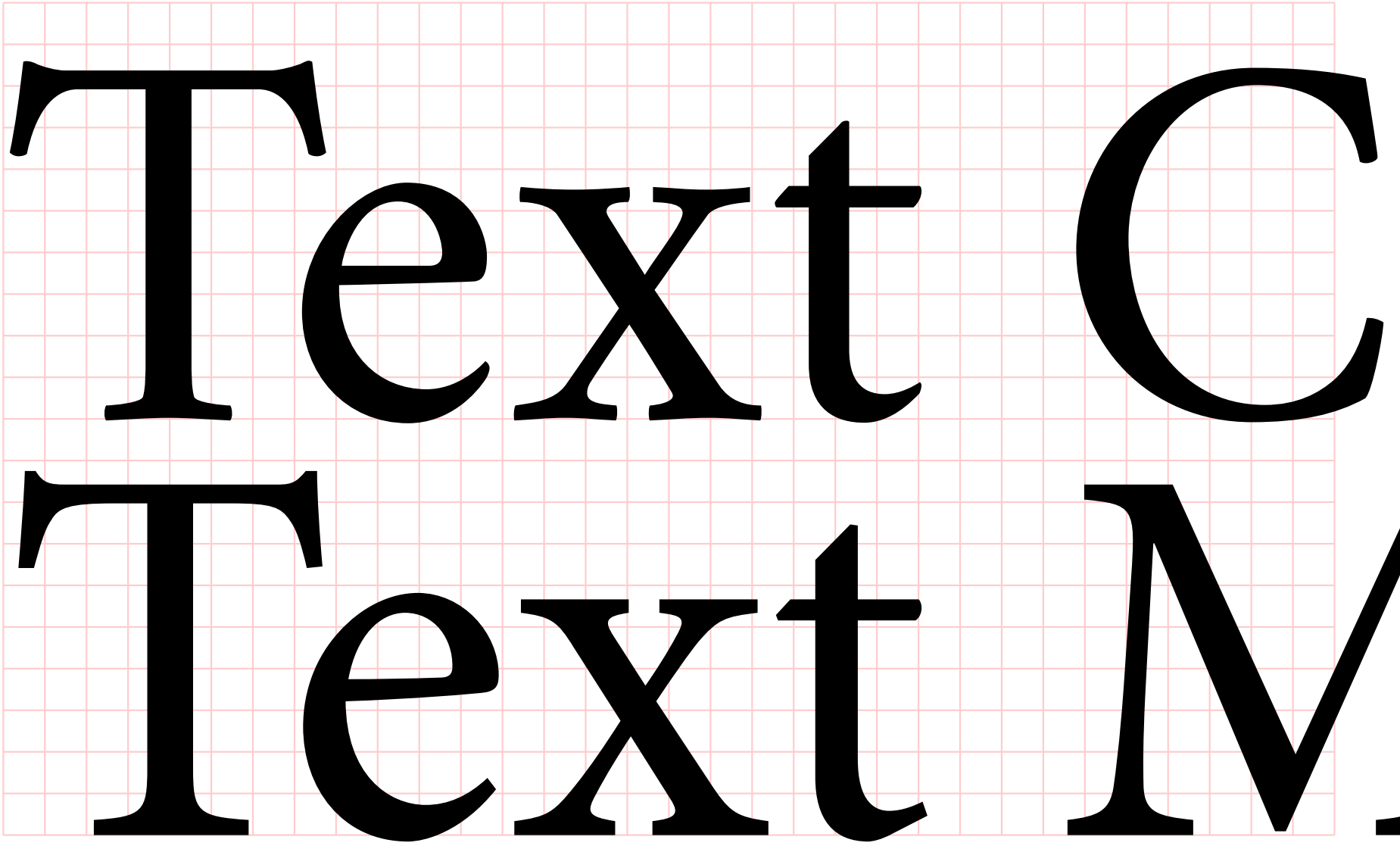
Cochineal

- an oldstyle text font family with Roman, Greek and Cyrillic alphabets
- derived from Sebastian Kosch's Crimson
- Roman alphabet resembles MinionPro but is not a metric clone thereof
- Regular, Italic, Bold, BoldItalic—otf and pfb
- roughly 1500 glyphs added to B ∪ I ∪ BI
- about 1500 glyphs now in each style
- available in OT1, T1, TS1, LY1, LGR, T2A, OT2 encodings
- used for body text in these slides

Cochineal

- the `.otf` files have much-expanded lookup tables
- figure styles {lining, osf} × {tabular, prop}, some of which I added (note the light, tight braces)
- superior and inferior figures in all four styles: e.g., SO_4
- superior roman letters, $^{\text{Abc123}}$
- small caps in all four styles: Small Caps, **Small Caps**, *Small Caps*, ***Small Caps***
- swash Q may be specified universally with a package option or individually with a macro: Q

Cochineal vs. MinionPro



Cochineal Greek and Cyrillic

- Greek is available in monotonic, polytonic and some ancient forms.
- Under LaTeX, Greek is available in LGR, used mainly by scholars who need to be able to generate short segments of polytonic and ancient Greek with a Western keyboard.
- Under LaTeX, Cyrillic is available in OT2, used mainly by scholars who need to be able to generate short segments of Cyrillic from a Western keyboard, and in T2A.
- Είναι όλα ελληνικά για μένα.
- Мое знание русского языка являеця жалким.

Cochineal issues

- as generated by autoinst (a wrapper for otftotfm) the word-spacing fontdimens (2, 3, 4, 7) are lower than specified in Cochineal's OTF TeX parameters
- must specify otftotfm's space-factor to get correct values
- this is a common problem with LaTeX support files generated by autoinst/otftotfm
- a font with 6000 glyphs is bound to have bugs, as its name suggests, especially with spacing and kerning

Cochineal Math based on newtxmath

Simplest form of the *Central Limit Theorem*: *Let* $X_1, X_2, \cdots$ *be a sequence of* i.i.d. *random variables with mean* 0 *and variance* 1 *on a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$. *Then*

$$\mathbb{P}\left(\frac{X_1 + \cdots + X_n}{\sqrt{n}} \le y\right) \xrightarrow[n\to\infty]{} \mathfrak{N}(y) := \int_{-\infty}^{y} \frac{\mathrm{e}^{-t^2/2}}{\sqrt{2\pi}}\,\mathrm{d}t,$$

or, equivalently, letting $S_n := \sum_1^n X_k$,

$$\mathbb{E}f\left(S_n/\sqrt{n}\right) \xrightarrow[n\to\infty]{} \int_{-\infty}^{\infty} f(t)\frac{\mathrm{e}^{-t^2/2}}{\sqrt{2\pi}}\,\mathrm{d}t, \qquad f \in \mathrm{b}\mathscr{C}(\mathbb{R}).$$

Nimbus15:

- derived from Nimbus fonts, metric clones of Courier, Helvetica and Times, issued in 2015 by URW++ by way of Artifex, makers of Ghostscript
- latest versions in update to the gs sources in October, 2015. Included in TeX Live 2016 in PostScript binary format, but without .afm files
- all now have Greek and Cyrillic alphabets
- license is incompatible with versions issued prior to 2000, on which TeXGyre fonts were based

NIMBUS15 MONO (COURIER)

- `NimbusMono-Regular->zco-Light`
  `NimbusMono-Bold->zco-Bold`
  `NimbusMono-Oblique->zco-LightOblique`
  `NimbusMono-BoldOblique->zco-BoldOblique`
- a new weight, intermediate between `Light` and **`Bold`**, was created with names `zco-Regular`, *`zco-Oblique`*
- glyphs in `Light`, `Regular` and **`Bold`** have stem widths 41em, 64em and 100em respectively
- the stem width in cmtt10 is 69em, slightly more than zco-Regular, its advance width is 525em, less than zco-Regular at 600em

Nimbus15 Mono Greek

- Greek glyphs support only monotonic Greek typography
- alpha (less fish-like) $\alpha \rightarrow \alpha$
- nu (curved, not v-shaped) $\nu \rightarrow \nu$
- Phi (less tall) $\Phi \rightarrow \Phi$

Nimbus15 Mono Narrow

- `zco-Regular` was modified to a narrow version, `zcoN-Regular`, starting with some FontForge Style/Change glyph transformations and then manually shortening serifs where necessary and making roundish glyph outlines narrower
- SAMPLE:

  `This is NimbusMonoNarrow, available only in regular weight, upright and` *`oblique`*`, advance width 500em, stem width 64em. IMO, it's not all that bad for rendering code segments.`

Nimbus15 Sans

- `NimbusSanL`, a metric clone of Helvetica, has been extended to include Greek (monotonic only) and Cyrillic glyphs.
- I changed the tonos accent from vertical to slanted for consistency with the Courier and Times clones.
- Given that TeX Gyre Heros has much more extensive coverage of Latin glyphs, the only usage that makes sense to me is for standalone Greek and Cyrillic.

Nimbus15 Serif

- `NimbusRomNo9L`, a metric clone of Times, has been extended by URW++ to include Greek (monotonic only) and Cyrillic glyphs
- current distribution from URW/Artifex has many gross errors in spacing and kerning of Greek and Cyrillic glyphs
- I expanded the Greek section so that polytonic and some ancient Greek forms are available, added a number of Cyrillic glyphs and tried to correct the spacing and kerning
- given that TeX Gyre Termes has much more extensive coverage of Latin glyphs, the only usage that makes sense to me is for standalone Gr/Cyr

LibertinusT1Math

- `Libertinus` is Khaled Hosny's fork of Libertine (`otf` only) with corrections and an added math font—`LibertinusMath` based on math symbols from Libertine
- he added many extendible symbols designed for use with unicode math
- `LibertinusT1Math` is my reworking of that math font into a LaTeX math package to accompany Libertine/Libertinus text
- Roman and Greek math letters are drawn from `Libertinus`

LibertinusT1Math conversion

- STIX LaTeX contains `.pl` files for all `.tfm` files—they are quite complete, with all glyph names included, providing a method for constructing STIX encoding files—these and `stix.sty` formed the basis for the construction of the encoding files and `.sty` file for `LibertinusT1Math`
- the glyph names were not the same in many cases—an expected complication
- unlike STIX math, which has its own calligraphic, BlackboardBold, script and gothic alphabets, `LibertinusT1Math` has only BB, and that may not be to everyones' taste, so I dropped the STIX based encodings based on those alphabets

LibertinusT1Math conversion [2]

- at the time I made these conversions, I did not see how to convert all information in the `otf` math tables to human readable form without much manual labor—I was able to get all the Top Accent and Italic Corrections by parsing the `.sfd`
- it turns out that the python program `ttx`, as of version 3.0, does provide this information and would have simplified this part of the project—thanks to KB for pointing this out
- extendible symbols designed for use with unicode math do not work properly with LaTeX math—making a proper math extension font was one of the more time-consuming parts of the project

LibertinusT1Math conversion [3]

- it turned out that many of the math symbols from Libertine were not horizontally aligned as they should have been (e.g., horizontal arrows) and had to be corrected so as to be centered on the math axis at `253em`
- horizontally extensible glyphs (e.g., overbraces) were constructed, as usual in LaTeX, in the `.sty` file
- in unicode math, this can be handled by code in the `.otf` math table
- using this math font will require in many cases the addition of separate math alphabets—the `mathalfa` package is set up to do this in a convenient way

LibertinusMath issues

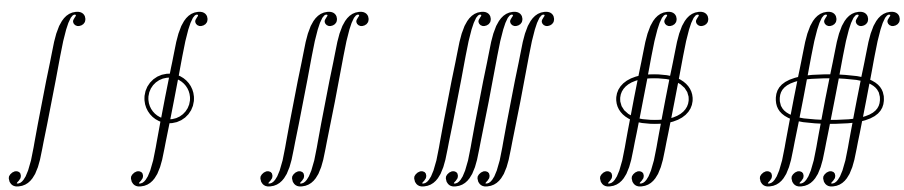
- upright integrals only—I added slanted versions as an option
- brace math delimiters seemed overly tight and a bit light—I changed them
- Libertine italic v and Greek nu are similar: $v \sim \nu$ Libertine's rounded v is used instead in LibertinusT1Math
- the binary relation symbols in Libertine seem rather small for old eyes

  $A \leq B, x \approx y$ (LibertinusMath)

  $A \leq B, x \approx y$ (newtxmath)

Newtxmath/newpxmath additions

- these math packages have been using an integral sign that is not to everyones' taste. I reworked it into an upright shape, much less wide, in 12 variants, 3 sizes and 2 weights, and from this produced a slanted form of each, so 144 new glyphs: e.g.,



(+6 more)

- following complaints about the overly tall large operators (e.g., `sum`, `product`), I constructed 76 new glyphs about 20% shorter at display size and about 12% shorter at text size, with selection controlled by the option `shorterops`

Newtxmath/newpxmath additions

- constructions using math delimiters with, e.g., `\biggl(`, were not producing the traditional sizes of output, and I reworked all the math delimiter glyphs to correct this