font-change-xetex

Version 2016.1

Macros to use OpenType and TrueType fonts
with XeTeX

Amit Raj Dhawan
amitrajdhawan@gmail.com

April 07, 2016

TABLE OF CONTENTS

1 Introduction

XƎTEX is a typesetting system based on TEX [1] that supports Unicode [2] and modern fonts like True Type fonts (TTF), Open Type fonts (OTF), and Apple Advanced Typography (AAT) fonts [3]. The possibility of using almost any kind of font format—either installed as a system font or just placed any TEX working-directory—has significantly improved the utility of TEX. Moreover, the incorporation of Unicode encoding in TEX documents is a progressive step as non-Latin scripts can now easily exploit the power of TEX. Variants of TEX like XƎTEX, XƎLATEX, and LuaTEX have made this feasible.

This document will describe the features of the package `font-change-xetex`. The package provides a simple way to use system-installed TrueType or OpenType fonts with XƎTEX. The package `fontspec` [4], which works with XƎLATEX and LuaTEX, is a very good tool for using TrueType and OpenType fonts, but to my knowledge, a similar package has not been proposed for XƎTEX. Following the "plain TEX" way of working, the macro package `font-change-xetex` allows users to employ *external* fonts with XƎTEX. The macro has been designed to change the text fonts in XƎTEX documents but one definition of the macro can be used to change some math mode fonts. The text fonts called by `font-change-xetex` can be used with the math fonts declared by another package called `font-change` [5] to obtain harmonious text and math font combinations.

I have been using `font-change-xetex` to typeset documents in English and Hindi from the year 2008, and from 2010, the macro has remained almost unchanged. The macro has worked with all the versions of XƎTEX so far, and it works with most font families installed on a computer system. Rarely, due to some font file issues, these fonts might not work properly but this is not a problem of the macro as such.

The macro `font-change-xetex` has been tested with the `xdvipdfmx` driver. Please pay attention to driver-related issues.

2 Using font-change-xetex

The package `font-change-xetex` can be downloaded from CTAN and it will be included in the future versions of MiKTEX and TEXLive. If `font-change-xetex` is installed in the TEX installation on your system, then it can be invoked by:

```
\input font-change-xetex
```

If the package is not installed in the TEX directory, the package file `font-change-xetex.tex` can be saved on the computer system, say in `C:/`, and can be invoked by typing the following in the `.tex` source file:

```
\input C:/font-change-xetex.tex
```

Once the package has been invoked, the commands included in it can be used. The following sections will discuss the font-changing commands of `font-change-xetex`.

3 Text font selection

The main font (font family) of a XƎTEX document can be changed by using the following definition:

`\myzfont{`*font name*`}{`*font size in points*`}{`*optional font features*`}`

where *font name* is the name of the installed font family (e.g., Warnock Pro, Minion Pro, Linux Libertine, etc.) or an individual font file (e.g., WarnockPro-Regular, WarnockPro-Semibold, etc.), *font size in points* is the size of the font in points stated as a positive integer without `pt` at the end (e.g., 9, 10, 31, etc.), and *optional font features* are declared to use or suppress advanced font features like ligatures, proportional figures, etc. (some of them will be discussed in the following section).

The definition `\myzfont` can be declared anywhere in the document. It is recommended to declare it only once somewhere at the beginning of the TEX document—this implements a regular change of fonts without exhausting TEX's memory.[1] Once a font family is declared, the font weight and style commands like **boldface** (`\bf`), *italics* (`\it`), *slanted* (`\sl`), ***italic boldface*** (`\itbf`), ***slanted boldface*** (`\slbf`), Caps (`\caps`), **Caps in Bold** (`\capsbf`), *Slanted Caps* (`\capssl`), and ***Slanted Caps in Boldface*** (`\capsslbf`) automatically use the corresponding fonts. If the declared font family does not have all features, then the features of the preceding font are used.[2]

With respect to the base font size declared in `pt`, for each font weight and style, the definition `\myzfont` includes several pre-defined relative font sizes of 5, 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, and 20 pt. That is, if we declare a base font size of 10 pt by `\myzfont{Minion Pro}{10}{}`, we will obtain the sizes of 5, 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, and 20 pts, and if we declare a base font size of 20 pt by `\myzfont{Minion Pro}{20}{}`, we get pre-defined sizes of 10, 12, 14, 16, 18, 20, 22, 24, 28, 32, and 40 pts.

`font-change-xetex` also provides the choice of a sans serif font through the definition:

`\mysanzfont{`*font name*`}{`*font size in points*`}{`*optional font features*`}`

This definition provides the same font styles, weights, and sizes as `\myzfont`. The font commands are also the same but each command has a prefix `sans` like `\sansrm`, `\sansit`, `\sanstwelvebf`, etc. The next section will mention all the included commands.

To change the font locally, `font-change-xetex` provides the following macro:

`\myfont{`*font name*`}{`*font size in points*`}{`*optional font features*`}`

The above command invokes the declared font (or font family) at the mentioned size with the same font styles and weights as provided by `\myzfont`. This command does not provide size variants, thereby easing the burden on TEX's font memory.

3.1 Examples of use

The following definition selects Warnock Pro as the main document font family at 10 pt. The TEX code is given in `brown` on the left and the corresponding results are given on the right.

```
\myzfont{Warnock Pro}{10}{}
```

`\rm This is regular`	This is regular
`\bf This is bold`	**This is bold**
`\it This is italic`	*This is italic*
`\sl This is slanted`	*This is slanted*
`\itbf This is italic bold`	***This is italic bold***

---

[1] The definition `\myfont` provides a way to use multiple font families within the same document but only at the declared size. Though all font style and weight variants like **boldface**, *italics*, *slanted*, ***italic boldface***, ***slanted boldface***, Caps, **Caps in Bold**, *Slanted Caps*, and ***Slanted Caps in Boldface*** will be available. Still, if someone wants to use `\myzfont` multiple times in a document, please send me an email. I have not included the variant of `font-change-xetex` that allows multiple declarations of `\myzfont` and `\mysanzfont` because the present variant works better with hanging punctuation.

[2] This works even with custom definitions like `\itbf`, `\caps`, etc. if a macro from `font-change` is declared before `font-change-xetex`.

`\slbf This is slanted bold`	***This is slanted bold***
`\caps This is Caps`	This is Caps
`\capsbf This is Bold Caps`	**This is Bold Caps**
`\capssl This is Slanted Caps`	*This is Slanted Caps*
`\capsslbf This is Slanted Bold Caps`	***This is Slanted Bold Caps***

`\fiverm This is 5 pt`	This is 5 pt
`\sixbf This is 6 pt`	**This is 6 pt**
`\sevenit This is 7 pt`	*This is 7 pt*
`\eightsl This is 8 pt`	*This is 8 pt*
`\nineitbf This is 9 pt`	***This is 9 pt***
`\slbf This is 10 pt`	***This is 10 pt***
`\caps This is 11 pt`	This is 11 pt
`\capsbf This is 12 pt`	**This is 12 pt**
`\capssl This is 14 pt`	*This is 14 pt*
`\capsslbf This is 16 pt`	***This is 16 pt***
`\eighteenrm This is 18 pt`	This is 18 pt
`\twentybf This is 20 pt`	**This is 20 pt**

The following statement selects Calibri as the sans serif font at 11 pt. The TEX code in `brown`, after compilation, gives the result displayed on its right.

`\mysanzfont{Calibri}{11}{}`

`\sansrm This is regular`	This is regular
`\sansbf This is bold`	**This is bold**
`\sansit This is italic`	*This is italic*
`\sanssl This is slanted`	*This is slanted*
`\sansitbf This is italic bold`	***This is italic bold***
`\sansslbf This is slanted bold`	***This is slanted bold***
`\sanscaps This is Caps`	This is Caps
`\sanscapsbf This is Bold Caps`	**This is Bold Caps**
`\sanscapssl This is Slanted Caps`	*This is Slanted Caps*
`\sanscapsslbf This is Slanted Bold Caps`	***This is Slanted Bold Caps***

`\sansfiverm This is 5.5 pt`	This is 5.5 pt
`\sanssixbf This is 6.6 pt`	**This is 6.6 pt**
`\sanssevenit This is 7.7 pt`	*This is 7.7 pt*
`\sanseightsl This is 8.8 pt`	*This is 8.8 pt*
`\sansnineitbf This is 9.9 pt`	***This is 9.9 pt***
`\sansslbf This is 11 pt`	***This is 11 pt***
`\sanscaps This is 12.1 pt`	THIS IS 12.1 PT
`\sanscapsbf This is 13.2 pt`	**THIS IS 13.2 PT**
`\sanscapssl This is 15.4 pt`	*THIS IS 15.4 PT*
`\sanscapsslbf This is 17.6 pt`	***THIS IS 17.6 PT***
`\sanseighteenrm This is 19.8 pt`	This is 19.8 pt
`\sanstwentybf This is 22 pt`	**This is 22 pt**

```
\myzfont{Warnock Pro}{10}{+pnum:+lnum}
\mysanzfont{Calibri}{11}{+pnum:+lnum}
%
\noindent{\sanstwelvecapsbf Text font change in action}
\medskip
\noindent The command {\tt \char'134 myzfont} changes the main text body font family
to Warnock Pro at 10\,pt with the current font set to regular. This is {\bf bold} and
this is {\sixteenitbf italic bold at 16\,pt}. {\sansrm The {\sanscaps sans serif}
document font is set to Calibri at 11\,pt.} In both the cases we have invoked the {\it
proportional figures}~(pnum) and {\it lining figures}~(lnum) features.

We can change the font locally to {\myfont{Tangerine}{20}{} Tangerine font at 20\,pt}
or {\myfont{Chaparral Pro}{11.4}{} Chaparral Pro at 11.4\,pt (this is {\bf Chaparral
bold}).
```

↓↓↓

TEXT FONT CHANGE IN ACTION

The command `\myzfont` changes the main text body font family to Warnock Pro at 10 pt with the current font set to regular. This is **bold** and this is ***italic bold at 16 pt***. The SANS SERIF document font is set to Calibri at 11 pt. In both the cases we have invoked the *proportional figures* (pnum) and *lining figures* (lnum) features.

We can change the font locally to *Tangerine font at 20 pt* or Chaparral Pro at 11.4 pt (this is **Chaparral bold**).

4 Hanging punctuation

Utilizing the character protrusion capability of XƎTEX (also possible with plain TEX), the `\hangpun` command of `font-change-xetex` provides a way to employ hanging punctuation, where the punctuation marks fall out of the text margin. From a typographic perspective, the use of hanging punctuation is debatable. In my opinion, it leads to more harmonious text justification and more elegant typesetting. The following text sample includes the same text with and without hanging punctuation.

Without hanging punctuation

Some people are for using hanging punctuation, and some against. In the end, it is a matter of personal choice. He said, “I use hanging punctuation.” “But does it really lead to better text justification?” asked Joe.

One may argue that protruding punctuation marks draw unnecessary attention but this is questionable. You should try it to find it out yourself! Well, let’s see . . .

With hanging punctuation

Some people are for using hanging punctuation, and some against. In the end, it is a matter of personal choice. He said, “I use hanging punctuation.” “But does it really lead to better text justification?” asked Joe.

One may argue that protruding punctuation marks draw unnecessary attention but this is questionable. You should try it to find it out yourself! Well, let’s see . . .

To activate hanging punctuation, the command `\hangpun` has to be declared after declaring `\myzfont` or `\myfont`. It can be deactivated by `\nohangpun`.

5 The OpenType layout

Vector scalable fonts (which includes most computer fonts) consist of glyph outlines and font layout tables. The glyph outlines of most Windows fonts are either TrueType or PostScript, and these font files are either in TrueType format (`.ttf` extenstion) or OpenType format (`.otf` extension). Generally, most modern PostScript outline fonts are in `.otf` format. The font layout tables of all `.otf` fonts (e.g., Warnock Pro, Minion Pro, Linux Libertine O, Calluna, etc.) and many `.ttf` fonts (e.g., Calibri, Georgia, Constantia, etc.) use the OpenType layout. This means that the OpenType font features included in these `.otf` and `.ttf` fonts can be accessed via OpenType layout tags [6].

An OpenType layout tag is an ASCII character string of 4 bytes that is used to identify the writing scripts, language systems, features and baselines of the font. Using XƎTEX’s `\font` command, the macros included in the package `font-change-xetex` can declare these tags by incorporating them as *optional font features*. A knowledge of some of these tags can be quite useful because different fonts have different default features, which the user may want to change. For example, the main font used in this document, Warnock Pro, invokes oldstyle figures (745.12) by default but scientific texts demand lining figures (745.12). By using the tag `lnum`, which overrides oldstyle figures with lining figures, we can make the desired switch. There are four categories of OpenType layout tags: Script, Language, Feature, and Baseline.

5.1 Script tags

Script tags generally correspond to a Unicode script and allow the user to provide information about the writing script. This feature is of great value when using non-Latin writing scripts or typesetting a document using multiple scripts. The sample texts below have been written in Devanagari using the Adobe Devanagari font but in the former case, the Latin script tag (non-default for this font) has been used. Due to this, the Devanagari ligatures are missing in the first text and the Hindi is syntactically incorrect.

```
\myfont{Adobe Devanagari}{12}{:script=latn}
```

विद्या अर्जन का प्रयास

Some vowels are misplaced and the ligatures are missing.

```
\myfont{Adobe Devanagari}{12}{:script=deva}
```

विद्या अर्जन का प्रयास

All vowels and ligatures are well placed.

Generally, font files have a default writing script. For example, all the fonts used in this document, except the ones used to typeset Devanagari, use Latin (`latn`) as the default script. A complete list of script tags can be found here.

5.2 Language tags

Language tags decide how text written in a given script is presented. For example, both English and French typesetting use the same writing script but the presentation of text is language dependent. A list of language tags is available here.

5.3 Feature tags

Feature tags allow the user to choose between different glyph forms. For example, we can choose whether to use standard ligatures (like ffi) or not (ffi). A table of feature tags is given here, and their description on this page. To use a font feature which has a tag called `ftag` via `font-change-xetex`, we have to type `:+ftag` in the *optional font features*. Suppose that the same feature is a default feature of the font, then in order to deactivate this feature we have to type `:-ftag` in the *optional font features*. The following example clarifies this usage.

```
\myfont{Warnock Pro}{10}{}% Declares Warnock Pro font at 10pt with the default settings

The font Warnock Pro considers {\bf Ţh Ťh ff fi fl ffi ffl fj ffj Th} as {\it standard
ligatures} ({\tt liga}), and {\bf st ct sp} as {\it discretionary ligatures}~({\tt
dlig}). The default settings of this font allow standard ligatures and suppress
discretionary ligatures. We will deactivate standard ligatures and turn on discretionary
ligatures with

\myfont{Warnock Pro}{10}{:-liga:+dlig}
% Standard Ligatures deactivated and Discretionary Ligatures activated

Now the standard ligatures ({\bf Ţh Ťh ff fi fl ffi ffl fj ffj Th}) have been turned
off and discretionary ligatures ({\bf st ct sp}) have been turned on.
```

↓↓↓

The font Warnock Pro considers **Ţh Ťh ff fi fl ffi ffl fj ffj Th** as *standard ligatures* (`liga`), and **st ct sp** as *discretionary ligatures* (`dlig`). The default settings of this font allow standard ligatures and suppress discretionary ligatures. We will deactivate standard ligatures and turn on discretionary ligatures with

Now the standard ligatures (**Ţh Ťh ff fi fl ffi ffl fj ffj Th**) have been turned off and discretionary ligatures (**st ct sp**) have been turned on.

The following example shows how the statements of the `font-change-xetex` can be used within a paragraph and how `\myfont` is used to exploit some features of Bickham Script Pro font.

```
\myfont{Bickham Script Pro}{30}{}
% Declares Bickham Script Pro font at 30pt with default features

This final statement is Crucial.

\myfont{Bickham Script Pro}{30}{:-calt:-clig}
% Contextual Alternates and Contextual Ligatures deactivated

This final statement is Crucial.

{\myfont{Bickham Script Pro}{30}{:+swsh}
% Swash glyphs activated

This final statement is Crucial.

% Alternate available fonts are accessed to locally change some glyphs
{\myfont{Bickham Script Pro}{30}{+aalt}T}his
{\myfont{Bickham Script Pro}{30}{+aalt=2}f}inal
statemen{\myfont{Bickham Script Pro}{30}{+aalt=3}t} is Crucial.}
```

↓↓↓

This final statement is Crucial.

This final statement is Crucial.

This final statement is Crucial.

This final statement is Crucial.

5.4 Baseline tags

These convey information about the baseline and about writing modes like horizontal, vertical, and math. More details about these tags can be found here.

6 Other features

This section will describe some font features provided by XƎTeX.

6.1 Font options

When an installed font is declared using `font-change-xetex`, its size and style variant are selected as well. However, if the user would like to force a particular variant of a font family, the following arguments can be used:

/B	Selects the **bold** variant of the declared font.
/I	Selects the *italic* variant of the declared font.
/BI	Selects the ***bold italic*** or ***italic bold*** variant of the declared font.
/IB	Same as /BI.
/S = x	Uses the font version corresponding to the optical size of x pt.
/AAT	only for Mac OS X.
/ICU	only for Mac OS X.

```
{\myfont{Warnock Pro/B}{10}{} sets Warnock Pro bold font as the regular font but it
is better to say} \myfont{Warnock Pro}{10}{} and then type {\bf some in bold} and {\rm
some in regular.}
```

↓↓↓

sets Warnock Pro bold font as the regular font but it is better to say and then type **some in bold** and some in regular.

6.1.1 Optical sizes

If a font family includes optical sizes then they can be employed using the `/S` option. Fonts with optical size variants produce finely adjusted typefaces according to type-size—this can improve text legibility and typesetting appeal. The default TeX font—Donald Knuth's Computer Modern font—has eight optical sizes: 5, 6, 7, 8, 9, 10, 12, and 17 pt. Because of these optical fonts, Computer Modern can typeset subscripts, sub-subscripts, superscripts, super-superscripts, and mathematics with elegance. The font used to typeset this document, Warnock Pro, offers four optical sizes: Caption, Regular, Subhead, and Display. Similar optical fonts are provided by other Adobe Pro fonts like Minion Pro, Chaparral Pro, Adobe Garamond Pro, etc.

When an installed font family is declared in XƎTeX, the optical sizes are chosen automatically. On my system (Windows 7 with XƎTeX version 0.99992), depending on the instructed font size, the Adobe Pro optical font variant is chosen automatically as:

Caption (<8.9 pt)	This is at 10 pt.
Regular (8.9 pt, 12.9 pt]	This is at 10 pt.
Subhead (12.9 pt, 22.9 pt]	This is at 10 pt.
Display (>22.9 pt)	This is at 10 pt.

If a font family supports optical sizes, then the definitions `\myzfont` and `\mysanzfont` select the appropriate sizes automatically. Using `\myfont`, we can force a particular optical font size (see the following example) as well. If the declared font family does not support optical sizes or if XƎTeX is not able to use them, then the usual font is used at the instructed size.

```
\twentyrm% Uses the 20 pt variant, that is, Subhead
\noindent Triangles-un-zybfi

\myfont{Warnock Pro/S=5}{20}{}% Forces the 5 pt variant, that is, Caption at 20 pt
\noindent Triangles-un-zybfi
```

↓↓↓

Triangles-un-zybfi
Triangles-un-zybfi

6.2 Color

In the *optional font features* of the definitions of `font-change-xetex`, the user can declare the font color using the argument:

`color=RRGGBB`

where `RR`, `GG`, and `BB` are the three respective pairs of Red, Green, and Blue hexadecimal values. Many RGB colors and their corresponding hexadecimal values can be found here.

```
\myfont{Warnock Pro}{10}{color=800080} This is somewhat purple.
```

↓↓↓

This is somewhat purple.

6.3 Letter-spacing

The letters of a declared font (installed on the system) can be spaced using the XƎTEX's `letterspace` option. The following example illustrates this.

```
\myfont{Warnock Pro}{10}{letterspace=10} The letters are too spaced.
```

↓↓↓

The letters are too spaced.

6.4 Inter-line and inter-word spacing

When a font is declared using `\myzfont`, `\mysanzfont`, or `\myfont` at a given size, the inter-line and inter-word spacing is done automatically depending on size of the font. This is done through the definition `\fontss` (declared in the macro `font-change-xetex`), which is:

```
\def\fontss{\parindent=2em%
\baselineskip=2.8ex%
\spaceskip=0.30001em plus0.11em minus0.11em}%
```

If other—small or large—font sizes are used via definitions like `\fourteenbf` or `\sevenrm`, it is recommended to declare `\fontss` immediately after declaring font-size commands. If one wants to change the parameters of the `\fontss` definition, a new definition with the same name but different parameters can be declared anywhere in the document—this will effect the following text. Also, it is possible to declare the just parameters included in `\fontss` differently.

```
\myzfont{Warnock Pro}{15}{} The type is large but the inter-line and inter-word space
have been taken into account.
\bigskip\bigskip
\twentyrm This is 30\,pt; the inter-line and inter-word spacing is not good.
\bigskip\bigskip
\twentyrm\fontss This is 30\,pt; the inter-line and inter-word spacing is better.
```

↓↓↓

The type is large but the inter-line and inter-word space have been taken into account.

This is 30 pt; the inter-line and inter-word spacing is not good.

This is 30 pt; the inter-line and inter-word spacing is better.

7 Special characters using Unicode

The package `font-change-xetex` includes several commands to typeset special Unicode characters. If the active font has the required glyph, then the following commands produce the corresponding special characters. Note that there is a prefix u before the character's common name.

`\uAlpha`	Α	`\uMu`	Μ	`\uPsi`	Ψ	`\ukappa`	κ
`\uBeta`	Β	`\uNu`	Ν	`\uOmega`	Ω	`\ulambda`	λ
`\uGamma`	Γ	`\uXi`	Ξ	`\ualpha`	α	`\umu`	μ
`\uDelta`	Δ	`\uOmicron`	Ο	`\ubeta`	β	`\unu`	ν
`\uEpsilon`	Ε	`\uPi`	Π	`\ugamma`	γ	`\uxi`	ξ
`\uZeta`	Ζ	`\uRho`	Ρ	`\udelta`	δ	`\uomicron`	ο
`\uEta`	Η	`\uSigma`	Σ	`\uepsilon`	ε	`\upi`	π
`\uTheta`	Θ	`\uTau`	Τ	`\uzeta`	ζ	`\urho`	ρ
`\uIota`	Ι	`\uUpsilon`	Υ	`\ueta`	η	`\usigmaf`	ς
`\uKappa`	Κ	`\uPhi`	Φ	`\utheta`	θ	`\usigma`	σ
`\uLambda`	Λ	`\uChi`	Χ	`\uiota`	ι	`\utau`	τ

\uupsilon	υ
\uphi	φ
\uchi	χ
\upsi	ψ
\uomega	ω
\uthetasym	ϑ
\uupsih	⍰
\upiv	ϖ
\ubull	•
\uhellip	…
\uprime	′
\uPrime	″
\uoline	‾
\ufrasl	⁄
\uweierp	℘
\uimage	ℑ
\ureal	ℜ
\utrade	™
\ualefsym	ℵ
\ularr	←
\uuarr	↑
\urarr	→
\udarr	↓
\uharr	↔
\ucrarr	↵
\ulArr	⇐
\uuArr	⇑
\urArr	⇒
\udArr	⇓
\uhArr	⇔
\uforall	∀
\upart	∂
\uexist	∃
\uempty	∅
\unabla	∇
\uisin	∈
\unotin	∉
\uni	∋
\uprod	∏
\usum	∑
\uminus	−
\ulowast	∗
\uradic	√
\uprop	∝
\uinfin	∞
\uang	∠
\uand	∧
\uor	∨
\ucap	∩
\ucup	∪
\uint	∫
\uthere4	∴
\usim	∼
\ucong	≅
\uasymp	≈
\une	≠
\uequiv	≡
\ule	≤
\uge	≥
\usub	⊂
\usup	⊃
\unsub	⊄
\usube	⊆
\usupe	⊇
\uoplus	⊕
\uotimes	⊗
\uperp	⊥
\usdot	⋅
\ulceil	⌈
\urceil	⌉
\ulfloor	⌊
\urfloor	⌋
\ulang	〈
\urang	〉
\uloz	◊
\uspades	♠
\uclubs	⍰
\uhearts	⍰
\udiams	♦
\ufnof	ƒ
\uensp	
\uemsp	
\uthinsp	
\uzwnj	
\uzwj	
\ulrm	
\urlm	
\undash	–
\umdash	—
\ulsquo	‘
\ursquo	’
\usbquo	‚
\uldquo	“
\urdquo	”
\ubdquo	„
\udagger	†
\uDagger	‡
\upermil	‰
\ulsaquo	‹
\ursaquo	›
\ueuro	€
\ucirc	ˆ
\utilde	˜
\uOElig	Œ
\uoelig	œ
\uScaron	Š
\uscaron	š
\uYuml	Ÿ
\uquot	”
\uamp	&
\uapos	’
\ult	<
\ugt	>
\unbsp	
\uiexcl	¡
\ucent	¢
\upound	£
\ucurren	¤
\uyen	¥
\ubrvbar	¦
\usect	§
\uuml	¨
\ucopy	©
\uordf	ª
\ulaquo	«
\unot	¬
\ushy	
\ureg	®
\umacr	¯
\udeg	°
\uplusmn	±
\usup2	²
\usup3	³
\uacute	´
\umicro	µ
\upara	¶
\umiddot	·
\ucedil	¸
\usup1	¹
\uordm	º
\uraquo	»
\ufrac14	¼
\ufrac12	½
\ufrac34	¾
\uiquest	¿
\uAgrave	À
\uAacute	Á
\uAcirc	Â
\uAtilde	Ã
\uAuml	Ä
\uAring	Å
\uAElig	Æ
\uCcedil	Ç
\uEgrave	È
\uEacute	É
\uEcirc	Ê
\uEuml	Ë
\uIgrave	Ì
\uIacute	Í
\uIcirc	Î
\uIuml	Ï
\uETH	Ð
\uNtilde	Ñ
\uOgrave	Ò
\uOacute	Ó
\uOcirc	Ô
\uOtilde	Õ
\uOuml	Ö
\utimes	×
\uOslash	Ø
\uUgrave	Ù
\uUacute	Ú
\uUcirc	Û
\uUuml	Ü
\uYacute	Ý
\uTHORN	Þ
\uszlig	ß
\uagrave	à
\uaacute	á
\uacirc	â
\uatilde	ã
\uauml	ä
\uaring	å
\uaelig	æ
\uccedil	ç
\uegrave	è
\ueacute	é
\uecirc	ê

`\ueuml`	ë	`\untilde`	ñ	`\udivide`	÷	`\uyacute`	ý
`\uigrave`	ì	`\uograve`	ò	`\uoslash`	ø	`\uthorn`	þ
`\uiacute`	í	`\uoacute`	ó	`\uugrave`	ù	`\uyuml`	ÿ
`\uicirc`	î	`\uocirc`	ô	`\uuacute`	ú	`\uhbar`	ħ
`\uiuml`	ï	`\uotilde`	õ	`\uucirc`	û		
`\ueth`	ð	`\uouml`	ö	`\uuuml`	ü		

We used the font Cambria Math to typeset the special unicode characters above. Other fonts, such as the main font family used in this document can be used as well. Note that some fonts may not include many Unicode characters.

```
The life of \ualpha\ was made better after tilting and becoming {\it \ualpha}---the
tilt was about 15\udeg. Even small {\sevenbf \uGamma} and elder {\fourteenbf \uGamma}
want to be {\sevenitbf \uGamma} and {\fourteenitbf \uGamma}. But \umu\ prefers to stay
upright when used in $2$\,\umu l.
```

↓↓↓

The life of α was made better after tilting and becoming *α*—the tilt was about 15°. Even small **r** and elder **Γ** want to be ***r*** and ***Γ***. But μ prefers to stay upright when used in 2 μl.

8 Changing some math fonts

This feature changes some math fonts and is currently under development. The changed math fonts belong to *family* 0; this means that regular numerals (0–9), regular letters (a–z and A–Z), upright Greek capitals (Γ, Δ, Θ, Λ, Ξ, Π, Σ, ϒ, Φ, Ψ, Ω), and hat (ˆ) are changed. As a consequence of this change, numbers in text mode (e.g., `149.75`) and in math mode (e.g., `$149.75$`) will be typeset using the same font. The change can be implemented by:

`\mymathfont`{*font name*}{*font size in points*}

It is possible to use `\mymathfont` multiple times in the same document. However, before reusing `\mymathfont`, the font size has to be reset by using the macro `\resetfontsize`, which is included in `font-change-xetex`.

Warning 1: The macro `\mymathfont` should be declared after `\myzfont` or `\myfont`, and after `\hangpun` or `\nohangpun`. In other words, the selection of the main body font and the choice between using hanging punctuation or not has to be made before invoking `\mymathfont`. If the definition `\mymathfont` is invoked and the text font is selected multiple times using `\myfont`, then after each selection, the math font must be redeclared using `\mymathfont`. Otherwise, the capital upright Greek letters in math mode will be missing.

```
\mysanzfont{Calibri}{11}{:+pnum:+lnum}% To use Calibri as the sans serif font
\input font_kp% From the package font-change; math fonts compatible with Warnock
Pro
\myzfont{Warnock Pro}{10}{:+pnum:+lnum}% To use Warnock Pro as the main font
\mymathfont{Warnock Pro}{10}% To typeset some math using Warnock Pro
\centerline{\sanstwelvecapsbf Math Font Test}
\bigskip
This font is Warnock Pro. The year is 2016. Check the number 2016 below
$$
 \sin(x) + e^{xy} + \Gamma.
$$
In the above expression 2016, $\sin$, (, ), and \uGamma\ are in Warnock Pro, whereas
e, x, and y are in Kp font.
```

↓↓↓

MATH FONT TEST

This font is Warnock Pro. The year is 2016. Check the number 2016 below

$$2016\,\sin(x) + e^{xy} + \Gamma.$$

In the above expression 2016, sin, (,), and Γ are in Warnock Pro, whereas e, x, and y are in Kp font.

Warning 2: It may happen that the declared math font does not have Greek characters or there are some coding issues. As a result of this, the Greek letters will not be typeset! In the following example, we use the font Chaparral Pro for some maths but despite being a “Pro” font, it does not have Greek letter support.

```
\mysanzfont{Calibri}{11}{:+pnum:+lnum}% To use Calibri as the sans serif font
\myzfont{Chaparral Pro}{11}{:+pnum:+lnum}% To use Chaparral Pro as the main font
\mymathfont{Chaparral Pro}{11}% To typeset some math using Chaparral Pro
\centerline{\sanstwelvecapsbf Failed Math Font Test}
\bigskip
This font is Chaparral Pro. Take a look at the following equation:
$$
\Psi(x) = 173x^2 + 22\cos(x) + \Gamma(x)
$$
In the above expression 173, 22, (, ), and $\cos$ are in Chaparral Pro, while x and =
are in Kp font, but the Greek letters are missing.
```

↓↓↓

FAILED MATH FONT TEST

This font is Chaparral Pro. Take a look at the following equation:

$$\boxtimes(x) = 173x^2 + 22\cos(x) + \boxtimes(x)$$

In the above expression 173, 22, (,), and cos are in Chaparral Pro, while x and = are in Kp font, but the Greek letters are missing.

9 font-change-xetex with font-change

The text font changing capabilities of `font-change-xetex` can be combined with the package `font-change` [7] which can change both math and text fonts. A font-changing macro from `font-change` has to be declared before the definitions of `font-change-xetex` so that the math font is chosen from `font-change` and the text font is selected by `font-change-xetex`. The definition `\mymathfont` can be used to change some math fonts as well. The following example shows this.

```
\mysanzfont{Calibri}{11}{:+pnum:+lnum}% To use Calibri as the sans serif font
\input font_kp% From the package font-change; math fonts compatible with Chaparral
Pro
\myzfont{Minion Pro}{10}{:+pnum:+lnum}% To use Minion Pro as the main font
\mymathfont{Minion Pro}{10}% To use Minion Pro for some math
\centerline{\sanstwelvecapsbf Text \& Math Font Test}
\bigskip
The text font is Minion Pro, most math is in Kp font, and some math is in Minion Pro.
\TeX's Kp fonts go very well with many fonts like Warnock Pro, Adobe Garamond Pro,
Minion Pro, etc. The text typed below shows how harmonious the mix of Minion Pro and
Kp fonts is.
$$
\Psi(x) = 173x^2 + 22\cos(x) + \Gamma(x)
$$
```

↓↓↓

TEXT & MATH FONT TEST

The text font is Minion Pro, most math is in Kp font, and some math is in Minion Pro. TEX's Kp fonts go very well with many fonts like Warnock Pro, Adobe Garamond Pro, Minion Pro, etc. The text typed below shows how harmonious the mix of Minion Pro and Kp fonts is.

$$\Psi(x) = 173x^2 + 22\cos(x) + \Gamma(x)$$

The following sample texts display some combinations of math fonts (TEX fonts invoked using `font-change`) and text fonts (OTF or TTF fonts invoked using `font-change-xetex`).

Warnock Pro (text and some math) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x,y,\theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2+y^2}} \int_{-2}^{2} f(x,y)\,dx\,dy?$$

Minion Pro (text and some math) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x,y,\theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2+y^2}} \int_{-2}^{2} f(x,y)\,dx\,dy?$$

Adobe Caslon Pro (text) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x,y,\theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2+y^2}} \int_{-2}^{2} f(x,y)\,dx\,dy?$$

Adobe Garamond Pro (text) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x,y,\theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2+y^2}} \int_{-2}^{2} f(x,y)\,dx\,dy?$$

Bergamo (text) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x,y,\theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2+y^2}} \int_{-2}^{2} f(x,y)\,dx\,dy?$$

Calluna (text) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Borgia Pro(text) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Linux Libertine (text and some math) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Gentium Basic (text) + **Kp** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$(x, y, \theta) = 342\,(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Warnock Pro (text and some math) + **Palatino** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Adobe Caslon Pro (text) + **Palatino** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Calluna (text) + **Palatino** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Adobe Garamond Pro (text) + **Palatino** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Linux Libertine (text and some math) + **Palatino** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Chaparral Pro (text) + **Charter** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x, y, \theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2 + y^2}} \int_{-2}^{2} f(x, y)\,dx\,dy?$$

Grandesign Neue Serif (text) + **Charter** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x,y,\theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2+y^2}} \int_{-2}^{2} f(x,y)\,dx\,dy?$$

Libre Baskerville (text) + **New Century Schoolbook** (math)

The mathematical typefaces should bear very close resemblance to the text typefaces. When comparing typefaces for “compatibility” we should investigate the *x-height*, the *m-width*, and the stroke width. What do you think about

$$\Psi(x,y,\theta) = 342\,\Gamma(\theta) + \sin(xy) + \arcsin(\theta) + \frac{1}{\sqrt{x^2+y^2}} \int_{-2}^{2} f(x,y)\,dx\,dy?$$

Bibliography

[1] Comprehensive TEX Archive Network (CTAN), “XƎTEX—an extended variant of TEX for use with Unicode sources.” [Online]. Available: https://www.ctan.org/pkg/xetex

[2] Unicode, Inc., “What is Unicode?” [Online]. Available: http://unicode.org/standard/WhatIsUnicode.html

[3] T. W. Phinney, “TrueType, PostScript Type 1, & OpenType: What’s the difference?” 2001.

[4] W. Robertson and K. Hosny, “The fontspec package: Font selection for XƎTEX and LuaTEX.” [Online]. Available: https://www.ctan.org/pkg/fontspec?lang=en

[5] Comprehensive TEX Archive Network (CTAN), “font-change—macros to change text and mathematics fonts in plain TEX.” [Online]. Available: https://www.ctan.org/pkg/font-change?lang=en

[6] Microsoft Corporation, “OpenType Layout tag registry.” [Online]. Available: https://www.microsoft.com/typography/otspec/TTOREG.htm

[7] A. R. Dhawan, “Macros to change text & math fonts in TEX: 45 beautiful variants,” 2015. [Online]. Available: https://www.ctan.org/tex-archive/macros/plain/contrib/font-change?lang=en