

Kp-fonts: OpenType version

Daniel FLIPO

22nd November 2022

This bundle provides OpenType versions of Type1 Kp-fonts designed by Christophe Caignaert. See `Kpfonts-Doc-English.pdf` for the full documentation of the original Type1 fonts.

It is usable only with LuaTeX or XeTeX engines; it consists of sixteen Text OpenType fonts, a Roman family **KpRoman** in eight shapes and weights—*Regular, Italic, Bold, BoldItalic, Light, LightItalic, Semibold, SemiboldItalic*—, a Sans-Serif family **KpSans** and a TypeWriter family **KpMono**, each of them in four shapes and weights—*Regular, Italic, Bold* and *BoldItalic*— and five maths OpenType fonts **KpMath-Regular**, **KpMath-Bold**, **KpMath-Light**, **KpMath-Semibold** and **KpMath-Sans**.

KpRoman and **KpSans** families have small caps available in two sizes (SmallCaps and PetiteCaps), upper and lowercase digits (0123456789), ancient ligatures ct , st and Q a long-tailed capital Q. Superior and inferior digits and letters have been added to the OpenType **KpRoman** and **KpSans** fonts for footnotes' calls and abbreviations 1st, 2nd...

The support of text fonts covers Latin and Latin Extended A (U+0020 to U+017F) but neither IPA nor Greek nor Cyrillic alphabets.

For all maths fonts, Latin and Greek letters are available in Upright and Italic shapes, in Bold and Regular weights: $\alpha \beta \text{C} \Delta$, $\alpha \beta \text{C} \Delta$, $\mathbf{\alpha} \mathbf{\beta} \mathbf{C} \mathbf{\Delta}$, $\mathbf{\alpha} \mathbf{\beta} \mathbf{C} \mathbf{\Delta}$.

Blackboard Bold capitals are available in two shapes, Serif and Sans: `\mathbb{ABC}` prints either $\mathbb{A}\mathbb{B}\mathbb{C}$ (default) or $\mathbb{A}\mathbb{B}\mathbb{C}$ (option `[Style=bbsans]`) Commands `\mathcal{ABC}` and `\mathscr{ABC}` print respectively \mathcal{ABC} and \mathscr{ABC} while `\mathfrak{ABCabc}` prints $\mathfrak{A}\mathfrak{B}\mathfrak{Cabc}$.

File `unimath-kpfonts.pdf` shows the full list of Unicode maths symbols provided by Kp-fonts, compared with other common maths fonts. More symbols, specific to Kp-fonts, are listed in section 3.2.

A style file `kpfont-otf.sty` is provided to load Kp-fonts easily. It is derived from `kpfont.sty` but options differ.

Please beware of the *experimental* status of the current version (0.47).

All fonts are covered by OFL licence, style file and documentation are under LPPL-1.3 licence.

oldstylenums: provides lowercase digits as a default.

To get uppercase digits locally: `{\addfontfeature{Numbers=Lining} 123}`.

Examples, upright, italic, bold and bolditalic:

- 0123456789 !
- 0123456789 !
- 0123456789 !
- 0123456789 !

oldstyle: provides lowercase digits as a default, long-tailed Q (Quebec) and (for Roman and Sans-Serif fonts only) old style ligatures ‘ct’ and ‘st’.

Examples:

- *Quest* for an attractive font!
- *Quest* for an attractive font!
- **Quest** for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!
- *Quest* for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!

veryoldstyle: same as option `oldstyle` but the round ‘s’ is replaced by the long one ‘ſ’ unless it ends a word or is followed by *b*, *f* or *h*² and ancient ligatures *fi*, *fl*, *ft* are activated. Coding `\textit{some of Highlands’ mysterious castles...}` will print *some of Highlandſ’ mysterious castles...* which is correct.

The automatic substitution relies on the OpenType feature `StylisticSet=12`. A round ‘s’ (resp. long ‘ſ’) can be forced by coding `s=` or `\shorts{}` (resp. `f3` or `\longs{}`).

largesmallcaps: prints larger SMALL CAPS than the default ones (`PETITE CAPS`).

altfligs: prints alternative shapes for ligatures *fi*, *fl*, *ffi*, *ffl* instead of *fi*, *fl*, *ffi*, *ffl*.

germandbls: `\SS` prints *SS* instead of *ß* (capital *Eszett*), ditto for small/petite caps.

eurosym: replaces the Euro character (€) by the official symbol (€) (available in regular, italic, bold and bolditalic).

harfbuzz: switches `Renderer=Harfbuzz` for HBLuaTeX engine; up to version 0.34, this renderer was silently activated but seldom useful.

1.3 Options for maths fonts *only*

lightmath: uses *light* maths fonts.

bbsans: command `\mathbb` prints Sans-Serif Blackboard Bold capitals with Serif fonts too: compare \mathbb{C} , \mathbb{K} , \mathbb{N} , \mathbb{Q} , \mathbb{R} , \mathbb{Z} , with \mathbb{C} , \mathbb{K} , \mathbb{N} , \mathbb{Q} , \mathbb{R} , \mathbb{Z} (default).

frenchstyle: Latin uppercase letters and all Greek letters are printed upright, only lowercase Latin letters are printed in italics; this follows the French typographic usage.

2. Rules found on [wikipedia](#).

3. On Unix systems the Compose key can be used: Compose f s.

oldstylenumsmath: prints lowercase digits in maths (default is uppercase).

narrowiints: prints condensed repeated integrals :

$\int\int$ and $\int\int\int$ instead of \iint and \iiint (default).

partialup: the `\partial` symbol is printed upright ∂ instead of ∂ .

fancyReIm: commands `\Re` and `\Im` print \Re and \Im respectively instead of \Re and \Im .

tight: horizontal spaces tightened in maths mode (same settings as `fourier-otf`).

noDcommand: do not define `\D` to avoid incompatibilities with other packages.

Please note that the **mathcal** option has been deleted: commands `\mathcal{ABC}` and `\mathscr{ABC}` now print ABC and \mathscr{ABC} respectively when `kpfonts-otf.sty` is loaded.

2 Another way to load Kp-fonts

Loading Kp-fonts through `kpfonts-otf.sty` offers only a limited choice of options; the standard commands `\setmainfont`, `\setsansfont`, `\setmonofont`, `\setmathfont` offer much more flexibility.

On the other hand, `kpfonts-otf.sty` defines a lot of useful commands to access AMS and specific Kp-fonts symbols. Loading `kpfonts-otf` with the `symbols` option enables to get all these commands defined without loading any font:

```
\usepackage[symbols]{kpfonts-otf}
```

Please note that `unicode-math`⁴ (and `fontspec`) *are loaded* by this procedure, no need to do it again, unless specific options are required, then `unicode-math` has to be loaded *before* `kpfonts-otf`, f.i.:

```
\usepackage[math-style=ISO,bold-style=upright]{unicode-math}
\usepackage[symbols]{kpfonts-otf}
```

Then, it is up to the user to load Kp-fonts with whatever option he/she likes using commands

```
\set...font{font}[options].
```

For documents requiring no maths fonts, loading `fontspec` and using the `\set...font` commands is enough, no need to load `kpfonts-otf` at all.

2.1 Options for Text fonts

Here are the options available for Text Kp-fonts:

Numbers=Lowercase to get lowercase digits 1,2,3 instead of 1,2,3; the default is **Numbers=Lining**.

4. A careful reading of both manuals `unicode-math.pdf` and `fontspec.pdf` (available in all TeX distributions) is required in order to take full advantage of these packages.

SmallCapsFeatures = {Letters=SmallCaps} the `\textsc{}` command will print larger SMALL CAPS than the default PETITE CAPS.

The default setting⁵, is **SmallCapsFeatures = {Letters=PetiteCaps}**.

Ligatures=TeX (default) ' !` ?` -- --- print respectively ' ; ; - —.

Ligatures=Common (default) automatic ligatures ff ffi ffl fi fl (plus s=).

StylisticSet=1 provides an alternative for glyphs ffi ffl fi fl (ff is unchanged).

Ligatures=Required : adds ft and tt ligatures.

Ligatures=Rare adds ct and st ligatures.

Style=Swash to get the long-tailed capital Q: Queen, also in small caps (both sizes): QUEEN and QUEEN.

Style=Historic replaces any instance of 's' by the long variant 'f'. It is still possible to get a round 's' coding it as 's='; this option should no longer be used, it is superseded by **StylisticSet=12** below.

StylisticSet=12 has been described in option `veryoldstyle` p. 3: it operates a contextual substitution of round 's' by long 'f'. An alias `Style=autoLongs` is available if `kpfonts-otf.sty` has been loaded.

Ligatures=Historic switches specific ligatures for the long f: fi, fl, ft.

StylisticSet=2 : \SS prints SS instead of ß (capital *Eszett*), ditto for small/petite caps.

StylisticSet=3 replaces the Euro character (€) by the official symbol (€) (available in regular, italics, bold and bolditalic).

Options may be are chosen for each font, say:

```
\setmonofont{KpMono}[Numbers=Lowercase,Style=Historic]
```

but can also be shared by different typefaces:

```
\defaultfontfeatures+[KpRoman,KpSans,KpMono]{Numbers=Lowercase}
\defaultfontfeatures+[KpRoman,KpSans]{%
  Ligatures = Rare,
  Style = Swash,
  SmallCapsFeatures = {Letters=PetiteCaps},
}
\setmainfont{KpRoman}
\setsansfont{KpSans}
\setmonofont{KpMono}
```

Notes : 1. `\setmonofont{KpMono}`, `\setsansfont{KpSans}`, `\setmainfont{KpRoman}` rely on files `KpMono.fontspec`, `KpSans.fontspec` and `KpRoman.fontspec` installed by `Kpfonts`.

2. Note the + ending `\defaultfontfeatures+` : options are *added*, not overwriting any other (including those of `fontspec.cfg`).

5. Changed in v0.37 to match the original `kpfonts` package.

3. Options can be gathered: `Ligatures={Rare,Historic}` (with braces) is equivalent to `Ligatures=Rare` and `Ligatures=Historic`.

4. These options can also be switched on and off *locally* using `\addfontfeatures` inside a group, f.i. to print lowercase digits 1234576890 with a font loaded with option `Numbers=Lining`:

```
{\addfontfeatures{Numbers=Lowercase}1234576890}
```

Actually, a shortcut is available in this case: `\oldstylenums{1234576890}`.

5. With the `KpRoman`, it is possible to define two more weights *Light* and *Semibold* borrowed from `KpLight`:

```
\newfontfamily\KpLight{KpLight}[<same options as KpRoman>]
\newcommand*\ltseries{\KpLight}
\newcommand*\sbseries{\KpLight\bfseries}
\DeclareTextFontCommand{\textlt}{\ltseries}
\DeclareTextFontCommand{\textsb}{\sbseries}
```

These weights are then available through `\ltseries`, `\sbseries` commands to be used in a group or alternatively through one argument commands `\textlt{}` and `\textsb{}`.

With the `KpLight`, weights *Semibold* and *Extrabold* can be defined similarly.

2.2 Options for maths fonts

The following options can be passed either to `unicode-math`⁶ or to `\setmathfont{}`:

math-style = ISO, TeX (default), french, upright;

bold-style = ISO, TeX (default), upright;

partial = upright (default italic);

nabla = italic (default upright);

Seven ‘Style Variants’ are available with Kp-fonts, here are the first three:

Style=mathcal (+ss01) commands `\mathcal{}` and `\mathscr{}` print *ABC* instead of *ℳℳℳ* (default), see note below;

Style=bbsans (+ss02) `\mathbb{}` prints Sans-Serif Blackboard bold capitals *ABC* for Serif maths fonts `KpMath-Regular` and `KpMath-Light` instead of *ABC* ;

Style=narrowiints (+ss03) provides condensed repeated integrals: \iiint instead of \iiint (default).

Note: if you want commands `\mathcal{ABC}` and `\mathscr{ABC}` to print *ABC* and *ℳℳℳ* respectively, you can use `unicode-math`’s option range this way:

```
\setmathfont{KpMath-Regular}[options]
\setmathfont{KpMath-Regular}[range={cal,bfcal},StylisticSet=1]
```

Both lines are mandatory: the first one loads `KpMath` as usual while the second one modifies `\mathcal{}` command’s output.

6. See the manual `unicode-math.pdf`.

The next four tables present the other Style Variants available:

Table 1 – Style=leqslant (+ss04)

Command	Default	Variant
<code>\leq</code>	\leq	\leqslant
<code>\geq</code>	\geq	\geqslant
<code>\nleq</code>	$\not\leq$	$\not\leqslant$
<code>\ngeq</code>	$\not\geq$	$\not\geqslant$
<code>\leqq</code>	$\leq\leq$	$\leq\leqslant$
<code>\geqq</code>	$\geq\geq$	$\geq\geqslant$
<code>\eqless</code>	\lessdot	\lesseqgtr
<code>\eqgtr</code>	\gtrdot	\gtreqless
<code>\lesseqgtr</code>	\lesseqgtr	\lesseqgtr
<code>\gtreqless</code>	\gtreqless	\gtreqless
<code>\lesseqqgtr</code>	\lesseqqgtr	\lesseqqgtr
<code>\gtreqqless</code>	\gtreqqless	\gtreqqless

Table 2 – Style=smaller (+ss05)

Command	Default	Variant
<code>\mid</code>	$ $	$\! $
<code>\nmid</code>	$\! $	$\! $
<code>\parallel</code>	\parallel	\parallel
<code>\nparallel</code>	$\not\parallel$	$\not\parallel$
<code>\parallelslant</code>	\parallel	\parallel
<code>\nparallelslant</code>	$\not\parallel$	$\not\parallel$

Table 3 – Style=subsetneq (+ss06)

Command	Default	Variant
<code>\subsetneq</code>	\subsetneq	\subsetneq
<code>\supsetneq</code>	\supsetneq	\supsetneq
<code>\subsetneqq</code>	\subsetneqq	\subsetneqq
<code>\supsetneqq</code>	\supsetneqq	\supsetneqq

Table 4 – Style=parallelslant (+ss07)

Command	Default	Variant
<code>\parallel</code>	\parallel	\parallel
<code>\nparallel</code>	$\not\parallel$	$\not\parallel$
<code>\shortparallel</code>	\parallel	\parallel
<code>\nshortparallel</code>	$\not\parallel$	$\not\parallel$

Example: switching styles 4 (leqslant) and 6 (subsetneq) can be achieved coding either `\setmathfont{KpMath-Regular.otf}[StylisticSet={4,6}]` or `\setmathfont{KpMath-Regular.otf}[Style={leqslant,subsetneq}]` but this second syntax is available only if `kpfonts-otf.sty` has been loaded (eventually with the `symbols` option).

Table 5 on the following page shows the available ‘Glyphs Variants’:

Example: with `\setmathfont{KpMath-Regular.otf}[CharacterVariant={3,6}]` commands `\epsilon` and `\phi` print ϵ and ϕ instead of ϵ and ϕ . The same is true of course for all shapes and and weights (upright, bold, bolditalic, sans-derif, etc.): f.i. with `math-syle=french`, `\epsilon` and `\phi` print ϵ and ϕ (upright).

Note about `\hbar`: `unicode-math` defines `\hbar` as `\hslash` (U+210F) while `amsmath` provides two different glyphs (italic h with horizontal or diagonal stroke). `kpfonts-otf` now follows `unicode-math`; the italic h with horizontal stroke can be printed using `\hslash` or `\hbar` together with character variant `cv01` or with `\mathbar` (replacement for AMS’ command `\hbar`).

Table 5 – Glyphs Variants

	Default	Variant	Command
cv00	\Re \Im	\Re \Im	<code>\Re</code> <code>\Im</code>
cv01	\hbar	\hbar	<code>\hslash</code> or <code>\hbar</code>
cv02	\emptyset	\emptyset	<code>\emptyset</code>
cv03	ϵ	ϵ	<code>\epsilon</code>
cv04	κ	κ	<code>\kappa</code>
cv05	π	π	<code>\pi</code>
cv06	ϕ	ϕ	<code>\phi</code>
cv07	ρ	ρ	<code>\rho</code>
cv08	σ	σ	<code>\sigma</code>
cv09	θ	θ	<code>\theta</code>
cv10	Θ	Θ	<code>\Theta</code>

3 Specific commands defined in `kpfonts-otf.sty`

3.1 Integrals

Kp-fonts offers variants for integral symbols suitable for indefinite integrals, they are coded as `\varint`, `\variint`, `\variiint`, `\variiiiint` and `\varidotsint`. Compare $\int f(t) dt$ and $\int f(t) dt$ and also

$$\int f(t) dt \quad \text{and} \quad \int f(t) dt$$

`\D{...}` prints an upright ‘d’ and improves kernings around the differential element: `\displaystyle\varint f(t)\D{t}` prints $\int f(t) dt$.

3.2 Specific maths symbols

The next tables present symbols unavailable as Unicode characters, they are coded in Kp-fonts’ private zone.

<code>\mmapsto</code>	\mapsto	<code>\longmmapsto</code>	\mapsto
<code>\mmapsfrom</code>	\mapsfrom	<code>\longmmapsfrom</code>	\mapsfrom
<code>\Mmapsto</code>	\Mmapsto	<code>\Longmmapsto</code>	\Mmapsto
<code>\Mmapsfrom</code>	\Mmapsfrom	<code>\Longmmapsfrom</code>	\Mmapsfrom
<code>\leftrightdasharrow</code>	\leftrightarrow	<code>\leadsto</code>	\leadsto
<code>\boxright</code>	\boxrightarrow	<code>\boxleft</code>	\boxleftarrow
<code>\circleright</code>	\circrightarrow	<code>\circleleft</code>	\circleftarrow
<code>\Diamondright</code>	\diamondrightarrow	<code>\Diamondleft</code>	\diamondleftarrow
<code>\boxdotright</code>	\boxdotrightarrow	<code>\boxdotleft</code>	\boxdotleftarrow
<code>\circledotright</code>	\circledrightarrow	<code>\circledotleft</code>	\circledleftarrow
<code>\Diamonddotright</code>	\diamonddotrightarrow	<code>\Diamonddotleft</code>	\diamonddotleftarrow

<code>\boxRight</code>	$\square\Rightarrow$	<code>\boxLeft</code>	$\square\Leftarrow$
<code>\boxdotRight</code>	$\square\Rightarrow$	<code>\boxdotLeft</code>	$\square\Leftarrow$
<code>\DiamondRight</code>	$\diamond\Rightarrow$	<code>\DiamondLeft</code>	$\diamond\Leftarrow$
<code>\DiamonddotRight</code>	$\diamond\Rightarrow$	<code>\DiamonddotLeft</code>	$\diamond\Leftarrow$
<code>\multimapdot</code>	\bullet	<code>\multimapdotinv</code>	\bullet
<code>\multimapdotboth</code>	$\bullet\bullet$		
<code>\multimapbothvert</code>	\circ	<code>\multimapdotbothvert</code>	\circ
<code>\multimapdotbothAvert</code>	\circ	<code>\multimapdotbothBvert</code>	\circ
<code>\capplus</code>	$\cap+$	<code>\sqcupplus</code>	$\sqcup+$
<code>\parallelslant</code>	$//$	<code>\colonsim</code>	$:\sim$
<code>\parallelbackslant</code>	$\backslash\backslash$	<code>\Colonsim</code>	$::\sim$
<code>\eqqColon</code>	$::=$	<code>\Colondash</code>	$::-$
		<code>\dashColon</code>	$-::$
<code>\strictif</code>	ε	<code>\strictfi</code>	ε
<code>\circledvee</code>	\circledvee	<code>\circledwedge</code>	\circledwedge
<code>\openJoin</code>	\times	<code>\opentimes</code>	\times
<code>\lambdaslash</code>	λ	<code>\lambdabar</code>	λ
		<code>\Vdash</code>	\Vdash
		<code>\Wr</code>	\Wr

<code>\idotsint</code>	$\int\cdots\int$	<code>\int</code>	\int
<code>\ointclockwise</code>	\oint	<code>\varointctrlockwise</code>	\oint
<code>\oiintclockwise</code>	\oiint	<code>\oiintctrlockwise</code>	\oiint
<code>\varoiintclockwise</code>	\varoiint	<code>\varoiintctrlockwise</code>	\varoiint
<code>\oiiintclockwise</code>	\oiiint	<code>\oiiintctrlockwise</code>	\oiiint
<code>\varoiiintclockwise</code>	\varoiiint	<code>\varoiiintctrlockwise</code>	\varoiiint
<code>\sqiint</code>	\sqiint	<code>\sqiiint</code>	\sqiiint

The full list of Unicode symbols available with Kp-fonts is shown in file `unimath-kpfonts.pdf`.

3.3 Wide accents

— `\wideoverbar` and `\mathunderbar`⁷

$$\overline{x} \quad \overline{xy} \quad \overline{xyz} \quad \overline{A \cup B} \quad \overline{A \cup (B \cap C) \cup D} \quad \overline{m + n + p}$$

7. `\overline` and `\underline` are not font related, they are based on `\rule`.

— `\widehat` and `\widetilde`

\widehat{x} \widehat{xx} \widehat{xxx} \widehat{xxxx} \widehat{xxxxx} \widehat{xxxxxx} \widetilde{x} \widetilde{xx} \widetilde{xxx} \widetilde{xxxx} \widetilde{xxxxx} \widetilde{xxxxxx}

— `\widecheck` and `\widebreve`

\widecheck{x} \widecheck{xxx} \widecheck{xxxxx} \widebreve{x} \widebreve{xxx} \widebreve{xxxxx}

— `\overparen` and `\underparen`

\overparen{x} \overparen{xy} \overparen{xyz} $\overparen{A \cup B}$ $\overparen{A \cup (B \cap C) \cup D}$ $\overparen{\frac{2}{x+y}}$ $\overparen{a+b+\dots+z}$

\underparen{x} \underparen{xz} \underparen{xyz} $\underparen{\frac{x+z}{2}}$ $\underparen{\frac{a+b+\dots+z}{26}}$

— `\overbrace` and `\underbrace`

\overbrace{a} \overbrace{ab} \overbrace{abc} \overbrace{abcd} \overbrace{abcde} $\overbrace{a+b+c}^3$ $\overbrace{a+b+\dots+z}^{26}$

\underbrace{a} \underbrace{ab} \underbrace{abc} \underbrace{abcd} \underbrace{abcde} $\underbrace{a+b+c}_3$ $\underbrace{a+b+\dots+z}_{26}$

— `\overrightarrow` and `\overleftarrow`

\overrightarrow{v} \overrightarrow{M} \overrightarrow{vv} \overrightarrow{AB} \overrightarrow{ABC} \overrightarrow{ABCD} $\overrightarrow{ABCDEFGH}$.

\overleftarrow{v} \overleftarrow{M} \overleftarrow{vv} \overleftarrow{AB} \overleftarrow{ABC} \overleftarrow{ABCD} $\overleftarrow{ABCDEFGH}$

— Enfin `\widearc` and `\widearcarrow` (or `\overrightarrowarc`)

\widearc{AMB} \widearcarrow{AMB}

3.4 Maths Versions

Different versions of the **KpMath** fonts may be defined in the document's preamble:

`\setmathfont{KpMath-Regular.otf}[version=base, options]`

`\setmathfont{KpMath-Bold.otf}[version=bold, options]`

`\setmathfont{KpMath-Semibold.otf}[version=semibold, options]`

`\setmathfont{KpMath-Sans.otf}[version=sans, options]`

`\setmathfont{KpMath-Light.otf}[version=light, options]`

then, it is easy to switch from one version to another one with `\mathversion{name}`.

Example ⁸ :

```
\setmathfont{KpMath-Regular.otf}[Style=leqslant, CharacterVariant=3]
\setmathfont{KpMath-Bold.otf}[version=bold,
    Style=leqslant, CharacterVariant=3]
\setmathfont{KpMath-Sans.otf}[version=sans,
    Style=leqslant, CharacterVariant=3]
```

Here is the same equation in three versions, normal, bold and Sans-Serif:

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{bold}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{sans}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

8. Option `CharacterVariant=3` changes ϵ into ε .