

Kp-fonts: OpenType version

Daniel FLIPO

8th August 2026

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 six maths OpenType fonts **KpMath-Regular**, **KpMath-Bold**, **KpMath-Light**, **KpMath-Semibold**, **KpMath-Sans** and **KpMath-SansBold**.

KpRoman and **KpSans** families have small caps available in two sizes (SmallCaps and PetiteCaps), upper and lowercase digits (0123456789), ancient ligatures *ct*, *st* and *Qa* 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: α β Γ Δ , α β Γ Δ , α β Γ Δ , α β Γ Δ .

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 (0.74).

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

1 Loading kpfonts-otf.sty

For users of the original kpfonts.sty package, the easiest way to try the OpenType version is to load kpfonts-otf.sty:

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

this loads unicode-math, fontspec and usually realscripts¹, and defines **KpRoman** (Regular or Light depending on options), **KpSans** and **KpMono** as Text fonts, **KpMath** (Regular or Light depending on options) as maths fonts.

kpfonts-otf.sty also defines all symbols available in latexsym and amssymb under the same names² and a bunch of Kp-fonts specifics symbols.

1.1 Global options for both text and maths

light: switches to *light* fonts, metrics are unchanged; *light* fonts might not look perfect on screen but they print fine.

Please compare *regular* (left) and *light* fonts (right):

Normal or light? Just a matter of taste. $E = mc^2$	Normal or light? Just a matter of taste. $E = mc^2$
--	--

<i>Normal or light? Just a matter of taste</i>	<i>Normal or light? Just a matter of taste</i>
--	--

Normal or light? Just a matter of taste	Normal or light? Just a matter of taste
--	--

<i>Normal or light? Just a matter of taste</i>	<i>Normal or light? Just a matter of taste</i>
---	---

sans: switches to *sans-serif* fonts for text and maths.

nomath: load neither unicode-math nor **KpMath** fonts; useful for documents without maths, or to choose other maths fonts.

notext: do not change any Text font, use the defaults.

nosf: do not change Sans-Serif Text fonts, use the defaults.

nott: do not change Typewriter Text fonts, use the defaults.

onlyrm: equivalent to the last two combined.

fulloldstyle: equivalent to options oldstyle and oldstylemath.

fulloldstylenums: equivalent to options oldstylenums and oldstylenumsmath.

¹See option fakedscripts below.

²Unicode names often differ from AMS names.

1.2 Options for text fonts *only*

lighttext: switches to *light* Text fonts.

Two more weights are provided by `kpfonts-otf.sty`: with the *lighttext* (or *light*) option, *Semibold* and *Extrabold* vs *Light* and *Semibold* without it. These weights are available through `\ltseries`, `\sbseries` and `\ebseries` commands to be used in a group or alternatively through one argument commands `\textlt{}`, `\textsb{}` and `\texteb{}`.

`{\sbseries\itshape Foo}` or `\textsb{\textit{Foo}}` print *Foo*.

sfstext: switches to *sans-serif* text fonts.

fakedscripts : prevents `realscripts` to be loaded, super- and sub-scripts will be produced by scaling (this is the way `pdflatex` works) which gives access to all characters available in the font but produces too tall and too thin glyphs. On the other hand `realscripts` gives access to the super/sub-scripts available in the font (a reduced set made of digits, lowercase latin letters, parentheses, etc. for the Kpfonts) which are better looking. I would recommend to keep `realscripts` loaded and to use the starred variants `\textsuperscript*`, `\textsubscript*`, to produce faked super/sub-scripts when needed, coding `H\textsuperscript*{\#}` to get $H^{\#}$ while `H\textsuperscript{\#}` would output $H^{\#}$ (# missing as superscript).

oldstylenums: provides lowercase digits as a default.

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

Examples, upright, italic, bold and bolditalic:

- `o123456789!`
- `o123456789!`
- `o123456789!`
- `o123456789!`

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

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 ‘f’ unless it ends a word or is followed by *b*, *f* or *h*³ and ancient ligatures *fi*, *fl*, *ft* are activated. Coding `\textit{some of Highlands’ mysterious castles...}` will print *some of Highlandf’ mysteriouf castlef...* which is correct.

The automatic substitution relies on the OpenType feature `StylisticSet=12`. A round ‘s’ (resp. long ‘f’) can be forced by coding `s=` or `\shorts{}` (resp. `f`⁴ 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.

sfmath: uses *sans-serif* 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.

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

narrowiints: prints condensed repeated integrals :

$\int\!\!\int$ and $\int\!\!\int\!\!\int$ instead of $\int\!\!\int$ and $\int\!\!\int\!\!\int$ (default).

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

fancyReIm: commands `\Re` and `\Im` print $\Re$ and $\Im$ respectively instead of $\mathbb{R}$ and $\mathbb{I}$.

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

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

³Rules found on [wikipedia](https://en.wikipedia.org/wiki/TeX_fonts).

⁴On Unix systems the Compose key can be used: Compose f s.

Please note that the **mathcal** option has been deleted: commands $\backslash\mathrm{mathcal}\{ABC\}$ and $\backslash\mathrm{mathscr}\{ABC\}$ now print ABC and $\mathcal{ABC}$ 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`⁵ (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**.

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

The default setting⁶, is **SmallCapsFeatures = {Letters=PetiteCaps}**.

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

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

⁵A 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.

⁶Changed in v0.37 to match the original `kpfonts` package.

StylisticSet=1 provides an alternative for glyphs ffi ffi 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. 4: 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 `Kp-fonts`, to define Italic, Bold, BoldItalic faces, there is no need for options `ItalicFont =` , `BoldFont =` .

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

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 1234567890 with a font loaded with option `Numbers=Lining`:

```
{\addfontfeatures{Numbers=Lowercase}1234567890}
```

Actually, a shortcut is available in this case: `\oldstylenums{1234567890}`.

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`⁷ 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 $\mathcal{ABC}$ instead of $\mathscr{ABC}$ (default), see note below;

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

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

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

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

⁷See the manual `unicode-math.pdf`.

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

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>	$\nleq$	$\nleqslant$
<code>\ngeq</code>	$\ngeq$	$\ngeqslant$
<code>\leqq</code>	$\leqq$	$\leqslant$
<code>\geqq</code>	$\geqq$	$\geqslant$
<code>\eqless</code>	$\lessdot$	$\lessgtr$
<code>\eqgtr</code>	$\gtrdot$	$\gtrless$
<code>\lesseqgtr</code>	$\lessgtr$	$\lesseqgtr$
<code>\gtreqless</code>	$\gtrless$	$\gtreqless$
<code>\lesseqqgtr</code>	$\lessgtr$	$\lesseqqgtr$
<code>\gtreqqless</code>	$\gtrless$	$\gtreqqless$
<code>\lesssim</code>	$\sim$	$\lesssim$
<code>\gtrsim</code>	$\sim$	$\gtrsim$

Table 2: Style=smaller (+ss05)

Command	Default	Variant
<code>\in</code>	$\in$	$\in$
<code>\ni</code>	$\ni$	$\ni$
<code>\setminus</code>	$\setminus$	$\setminus$
<code>\mid</code>	$\mid$	$\mid$
<code>\nmid</code>	$\nmid$	$\nmid$
<code>\parallel</code>	$\parallel$	$\parallel$
<code>\nparallel</code>	$\nparallel$	$\nparallel$
<code>\parallelslant</code>	$\parallel$	$\parallel$
<code>\nparallelslant</code>	$\nparallel$	$\nparallel$

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>	$\nparallel$	$\nparallel$
<code>\shortparallel</code>	$\parallel$	$\parallel$
<code>\nshortparallel</code>	$\nparallel$	$\nparallel$

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-style=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 `\mithbar`

Table 5: Glyphs Variants

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

(replacement for AMS' command $\backslash\mathrm{hbar}$).

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 $\backslash\mathrm{varint}$, $\backslash\mathrm{variint}$, $\backslash\mathrm{variiint}$, $\backslash\mathrm{variiiiint}$ and $\backslash\mathrm{varidotsint}$. Compare $\int f(t) dt$ and $\int f(t) dt$ and also

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

$\backslash\mathrm{D}\{\dots\}$ prints an upright 'd' and improves kernings around the differential element:
 $\backslash\mathrm{displaystyle}\backslash\mathrm{varint} f(t)\backslash\mathrm{D}\{t\}$ 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.

$\backslash\mathrm{mmapsto}$	$\mapsto$	$\backslash\mathrm{longmmapsto}$	$\mapsto$
$\backslash\mathrm{mmapsfrom}$	$\mapsfrom$	$\backslash\mathrm{longmmapsfrom}$	$\mapsfrom$
$\backslash\mathrm{Mmapsto}$	$\mapsto$	$\backslash\mathrm{Longmmapsto}$	$\mapsto$
$\backslash\mathrm{Mmapsfrom}$	$\mapsfrom$	$\backslash\mathrm{Longmmapsfrom}$	$\mapsfrom$
$\backslash\mathrm{leftrightrightarrow}$	$\leftrightarrow$	$\backslash\mathrm{leadsto}$	$\leadsto$

<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>	$\circrightarrow$	<code>\circledotleft</code>	$\circleftarrow$
<code>\Diamonddotright</code>	$\diamondrightarrow$	<code>\Diamonddotleft</code>	$\diamondleftarrow$
<code>\boxRight</code>	$\boxRightarrow$	<code>\boxLeft</code>	$\boxLeftarrow$
<code>\boxdotRight</code>	$\boxdotRightarrow$	<code>\boxdotLeft</code>	$\boxdotLeftarrow$
<code>\DiamondRight</code>	$\DiamondRightarrow$	<code>\DiamondLeft</code>	$\DiamondLeftarrow$
<code>\DiamonddotRight</code>	$\DiamondRightarrow$	<code>\DiamonddotLeft</code>	$\DiamondLeftarrow$
<code>\multimapdot</code>	$\multimap$	<code>\multimapdotinv</code>	$\multimap$
<code>\multimapdotboth</code>	$\multimap$		
<code>\multimapbothvert</code>	$\multimap$	<code>\multimapdotbothvert</code>	$\multimap$
<code>\multimapdotbothAvert</code>	$\multimap$	<code>\multimapdotbothBvert</code>	$\multimap$
<code>\capplus</code>	$\capplus$	<code>\sqcupplus</code>	$\sqcupplus$
<code>\parallelslant</code>	$\parallel$	<code>\colonsim</code>	$\colon\sim$
<code>\parallelbackslant</code>	$\parallel$	<code>\Colonsim</code>	$\Colon\sim$
<code>\eqqColon</code>	$\equiv$	<code>\Colondash</code>	$\Colon-$
<code>\strictif</code>	$\rightarrow$	<code>\strictfi</code>	$\leftarrow$
<code>\circledvee</code>	$\vee$	<code>\circledwedge</code>	$\wedge$
<code>\openJoin</code>	$\times$	<code>\opentimes</code>	$\times$
<code>\lambdash</code>	λ	<code>\lambdabar</code>	λ
<code>\idotsint</code>	$\int$		
<code>\ointclockwise</code>	$\oint$	<code>\varointctrclockwise</code>	$\oint$
<code>\oiintclockwise</code>	$\oiint$	<code>\oiintctrclockwise</code>	$\oiint$
<code>\varoiintclockwise</code>	$\varoiint$	<code>\varoiintctrclockwise</code>	$\varoiint$
<code>\oiintclockwise</code>	$\oiint$	<code>\oiintctrclockwise</code>	$\oiint$
<code>\varoiintclockwise</code>	$\varoiint$	<code>\varoiintctrclockwise</code>	$\varoiint$
<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`⁸

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

- `\widehat` and `\widetilde`

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

- `\widecheck` and `\widebreve`

$$\check{x} \quad \overline{xxxx} \quad \overline{xxxxxx} \quad \check{x} \quad \overline{xxxx} \quad \overline{xxxxxx}$$

- `\overparen` and `\underparen`

$$\begin{array}{ccccccc} \bar{x} & \bar{xy} & \overline{xyz} & \overline{A \overset{\circ}{\cup} B} & \overline{A \cup (B \cap C) \overset{\circ}{\cup} D} & \overset{2}{\overline{x+y}} & \overline{a+b+\dots+z}^{26} \\ & & & \underline{x} & \underline{xz} & \underline{xyz} & \underline{x+z} & \underline{a+b+\dots+z} \\ & & & & & & 2 & 26 \end{array}$$

- `\overbrace` and `\underbrace`

$$\begin{array}{ccccccccc} \overline{a} & \overline{ab} & \overline{abc} & \overline{abcd} & \overline{abcde} & \overbrace{a+b+c}^3 & \overbrace{a+b+\dots+z}^{26} \\ \underline{a} & \underline{ab} & \underline{abc} & \underline{abcd} & \underline{abcde} & \underbrace{a+b+c}_3 & \underbrace{a+b+\dots+z}_{26} \end{array}$$

- `\overrightarrow` and `\overleftarrow`

$$\begin{array}{ccccccc} \vec{v} & \vec{M} & \vec{vv} & \vec{AB} & \vec{ABC} & \vec{ABCD} & \vec{ABCDEFGH} \\ \overleftarrow{v} & \overleftarrow{M} & \overleftarrow{vv} & \overleftarrow{AB} & \overleftarrow{ABC} & \overleftarrow{ABCD} & \overleftarrow{ABCDEFGH} \end{array}$$

- Finally `\widearc` and `\widearccarrow` (or `\overrightarc`)

$$\overbrace{AMB} \quad \overrightarrow{AMB}$$

All the extensible arrows provided by the `mathtools` package are available (loading `kpfonts-otf.sty` is required), f.i.:

$$X \overset{\text{above}}{\rightleftharpoons} Y \underset{\text{under}}{\hookrightarrow} Z \overset{\text{above}}{\mapsto} W$$

A wide range of extensible vertical delimiters is provided:

$$\left/ \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right[\begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix}] \left\{ \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix} \right\} | a_1 | a_1 ||| a_1 ||| a_1 ||| \left[\begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix} \right] | a_1 | | a_1 | | \langle a_1 \rangle \langle a_1 \rangle \langle a_1 \rangle$$

⁸`\overline` and `\underline` are not font related, they are based on `\rule`.

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}`.

Example⁹:

```
\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:

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

`\mathversion{bold}`

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

`\mathversion{sans}`

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

To get bold formulas you do not need to define `\mathversion{bold}`, you can just use the `\boldmath` command: f.i. `\boldmath $E=mc^2$` prints **$E = mc^2$** .

⁹Option `CharacterVariant=3` changes ϵ into ε .