

# Kp-fonts version OpenType

Daniel FLIPO

16 mars 2021

Le but de ce travail est de proposer une version OpenType des polices Kp-fonts dessinées par Christophe Caignaert au format type1. Pour une présentation complète de celles-ci, on pourra consulter la documentation originale [Kpfonts-Doc-French.pdf](#).

La présente distribution, utilisable uniquement avec LuaTeX ou XeTeX, contient seize polices OpenType de texte, une famille romaine **KpRoman** en huit déclinaisons — *Regular, Italic, Bold, BoldItalic, Light, LightItalic, Semibold, SemiboldItalic* —, une famille sans-serif **KpSans** et une à espacement fixe **KpMono** chacune en quatre déclinaisons — *Regular, Italic, Bold* et *BoldItalic* — et cinq polices mathématiques **KpMath-Regular**, **KpMath-Bold**, **KpMath-Light**, **KpMath-Semibold** et **KpMath-Sans**.

Les familles **KpRoman** et **KpSans** ont de vraies petites capitales en deux tailles (NORMALE et GRANDE), des chiffres elzéviens (0123456789) et offrent des ligatures anciennes comme çt, st et le Q long. Les chiffres et lettres supérieures et inférieures ont été ajoutées aux polices OpenType pour les appels de notes et les abréviations M<sup>me</sup> 1<sup>er</sup>, etc.

Les cinq polices mathématiques disposent des lettres latines et grecques droites, italiques, grasses droites et grasses italiques :  $a \beta C \Delta$ ,  $a \beta C \Delta$ ,  $\mathbf{a} \beta C \Delta$ ,  $\mathbf{a} \beta C \Delta$ .

la commande `\mathbb{ABC}` affiche au choix  $\mathbb{ABC}$  (option `Style=bbsans`) ou  $\mathbb{ABC}$  par défaut, les commandes `\mathcal{ABC}` et `\mathscr{ABC}` affichent  $\mathcal{ABC}$  par défaut ou  $\mathscr{ABC}$  avec l'option `Style=mathcal`, tandis que la commande `\mathfrak{ABCabc}` affiche  $\mathfrak{ABCabc}$ .

La liste complète des symboles mathématiques présents dans Unicode et disponibles dans les polices **KpMath** est présentée dans le fichier [unimath-kpfonts.pdf](#). Des symboles supplémentaires sont également disponibles, ils sont présentés à la section 3.2.

La distribution est accompagnée d'un fichier `kpfont-otf.sty` destiné à faciliter la transition, certaines options de la version originale ont été reprises, d'autres ont été abandonnées...

Veuillez noter que la présente version (0.35) est *expérimentale*.

La distribution est proposée sous licence OFL pour les polices et LPPL-1.3 pour les fichiers LaTeX.

# 1 Chargement par `kpfonts-otf.sty`

Les utilisateurs de la version originale des Kp-fonts préféreront probablement charger `kpfonts-otf.sty` comme ceci :

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

ceci charge `unicode-math` et donc `fontspec` et définit `KpRoman` (`Regular` ou `Light` selon les options), `KpSans` et `KpMono` comme polices de texte et `KpMath` (`Regular` ou `Light` selon les options) comme police mathématique.

En plus, tous les symboles disponibles dans `latexsym` et `amssymb` sont définis sous les mêmes noms dans `kpfonts-otf.sty`<sup>1</sup> ainsi qu'un certain nombre de symboles et de commandes spécifiques à Kp-fonts.

## 1.1 Options globales affectant texte et maths

**light** : pour utiliser la version *light* des polices. Les métriques sont inchangées. L'affichage n'est pas très bon, au contraire de l'impression si vous aimez les polices fines.

Ce texte est composé avec les polices par défaut, vous pourrez comparer ci-dessous les versions *regular* (à gauche) et *light* (à droite) :

Normal ou léger? tous les goûts sont dans la nature... $E = mc^2$	Normal ou léger? tous les goûts sont dans la nature... $E = mc^2$
---	---

<i>Normal ou léger? tous les goûts sont dans la nature...</i>	<i>Normal ou léger? tous les goûts sont dans la nature...</i>
---	---

<b>Normal ou léger? tous les goûts sont dans la nature...</b>	<b>Normal ou léger? tous les goûts sont dans la nature...</b>
---	---

<i>Normal ou léger? tous les goûts sont dans la nature...</i>	<i>Normal ou léger? tous les goûts sont dans la nature...</i>
---	---

**nomath** : ne charge ni `unicode-math` ni les fontes `KpMath` ; utile pour les documents sans maths ou pour faire appel à d'autres fontes pour les maths.

**notext** : Kp-fonts ne modifie aucune des polices de texte par défaut.

**nosf** : Kp-fonts ne modifie pas les polices sans-serif de texte.

**nott** : Kp-fonts ne modifie pas les polices de texte à chasse fixe.

**onlyrm** : équivalente aux deux précédentes options.

## 1.2 Options des polices de texte

Notez que ces options n'affectent que les polices de texte.

**lighttext** : le texte est composé en utilisant les polices « light ».

---

1. Les noms Unicode officiels diffèrent souvent.

**oldstylenums** : Les polices fournissent ici les nombres elzéviens par défaut.

Pour accéder aux nombres usuels : `{\addfontfeature{Numbers=Lining} 123}` (à faire dans un groupe).

Quelques exemples en droit et italique, medium et gras :

— 0123456789!

— *0123456789!*

— **0123456789!**

— ***0123456789!***

**oldstyle** : Les polices fournissent ici les nombres elzéviens par défaut, le Q-long (Qui?), les ligatures anciennes « ct » et « st » (en romain et sans-serif uniquement).

Quelques exemples :

— *Quelle police, quelle esthétique actuelle!*

— *Quelle police, quelle esthétique actuelle!*

— **Quelle police, quelle esthétique actuelle!**

— **QUELLE POLICE, QUELLE ESTHÉTIQUE ACTUELLE!**

— *Quelle police, quelle esthétique actuelle!*

— **QUELLE POLICE, QUELLE ESTHÉTIQUE ACTUELLE!**

**veryoldstyle** : identique à l'option `oldstyle` mais en plus le « s » rond est remplacé par le « f » long sauf en fin de mot (où il doit rester rond) et les ligatures fi, fl, ft sont activées. Ceci fonctionne assez bien avec LuaTeX : le codage

`\textit{« Des espoirs si funestes... »}` donne « *Des espoirs si funestes...* ».

Les choses se passent bien tant que le « s » est immédiatement suivi d'une espace Unicode (U+0020 espace sécable de base, U+00A0 espace-mot insécable, U+202F fine insécable) ou d'un signe de ponctuation (virgule, point, etc.) *mais pas* s'il est suivi d'une espace insécable produite par TeX (sous la forme pénalité + espace)<sup>2</sup>!

Exemple : le codage `\textit{espoirs, mais quels~espoirs.}` produit *espoirs, mais quels espoirs?*

Même problème devant la ponctuation haute en français, sauf à utiliser l'option `UnicodeNoBreakSpaces` de `babel-french`. On doit forcer le « s » rond final en codant `s=: \textit{espoirs? désespoirs=!}` produit *espoirs? désespoirs!*

Avec XeTeX il faut forcer le « s » rond final avant toutes les espaces en codant `s=.`

**largesmallcaps** : produit des petites capitales plus GRANDES que par les PETITES obtenues par défaut.

**altfligs** : affiche des variantes des ligatures fi, fl, ffi, ffl au lieu de fi, fl, ffi, ffl.

**germandbls** : `\SS` affiche SS au lieu de ß (*eszett* majuscule), idem en petites capitales.

**eurosym** : remplace le caractère Euro (€) par le symbole officiel (€), disponible également en italique, gras et gras italique; la commande `\kpeuro` produit également ce symbole officiel, elle admet un argument optionnel : `\kpeuro[1,50]{}` imprime 1,50 € (espace-mot insécable).

**harfbuzz** : active le moteur de rendu `Renderer=Harfbuzz` pour HBLuaTeX; depuis la version 0.35, il n'est plus activé par défaut (il est rarement nécessaire).

---

2. On utilise la *feature* OpenType `calt` qui ne reconnaît pas les `\penalty` de TeX.

### 1.3 Options des polices mathématiques

**lightmath** : les mathématiques sont composées avec les polices « light ».

**bbsans** : dans tous les cas, la commande `\mathbb` produit du sans-serif :  $\mathbb{C}$ ,  $\mathbb{K}$ ,  $\mathbb{N}$ ,  $\mathbb{Q}$ ,  $\mathbb{R}$ ,  $\mathbb{Z}$ , à comparer à  $\mathbb{C}$ ,  $\mathbb{K}$ ,  $\mathbb{N}$ ,  $\mathbb{Q}$ ,  $\mathbb{R}$ ,  $\mathbb{Z}$ .

**mathcal** : cette option modifie l'effet des commandes `\mathcal` et `\mathscr`, elles produisent toutes les deux  $\mathcal{ABCD}$  alors que par défaut on a  $\mathscr{ABCD}$ .

**frenchstyle** : les majuscules romaines et toutes les lettres grecques (majuscules ou minuscules) sont droites, selon la tradition typographique française, seules les **minuscules** romaines restent obliques.

**oldstylenumsmath** : fournit les nombres elzéviens en mode mathématique.

**narrowiints** : fournit des intégrales multiples plus étroites, plus « françaises » :

$\int\int$  et  $\int\int\int$  au lieu de  $\iiint$  et  $\iiiii$

**partialup** : le symbole `\partial` affiche  $\partial$  (droit) au lieu de  $\partial$ .

**fancyReIm** : les commandes `\Re` et `\Im` donnent normalement  $\Re$  et  $\Im$ , avec cette option elles produisent respectivement  $\Re$  et  $\Im$ .

**tight** : les espaces horizontaux sont réduits en math (comme ceux de `fourier-otf`).

**noDcommand** : la commande `\D` n'est pas définie (risque d'incompatibilité).

## 2 Autre façon de charger les Kp-fonts

Lorsqu'on charge les Kp-fonts comme indiqué à la section précédente en utilisant `kpfonts-otf.sty`, le choix des options reste assez limité. Certains préféreront probablement utiliser directement les commandes `\setmainfont`, `\setsansfont`, `\setmonofont`, `\setmathfont` pour avoir accès à toutes leurs options. Parmi eux, ceux qui voudraient bénéficier des définitions de symboles mathématiques faites dans `kpfonts-otf`, peuvent charger celui-ci avec l'option `symbols` qui permet d'accéder à toutes ces définitions *sans charger aucune fonte* :

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

Noter que dans ce cas, `unicode-math`<sup>3</sup> (et `fontspec`) sont chargés par `kpfonts-otf`, inutile donc de les (re)charger, sauf à le faire *avant* l'appel à `kpfonts-otf` pour leur passer des options, par exemple :

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

L'utilisateur choisit ensuite librement ses polices de texte romaine, sans-serif et à chasse fixe ainsi que la police mathématique en utilisant la syntaxe classique

```
\set...font{nom-fonte}[options].
```

Ceux qui veulent utiliser les KP-fonts pour des documents sans maths, peuvent évidemment charger uniquement `fontspec` et faire appel aux commandes ci-dessous.

3. Une lecture attentive des manuels `unicode-math.pdf` et `fontspec.pdf` (disponibles dans toutes les distributions TeX) est indispensable pour bien maîtriser leur utilisation.

## 2.1 Options des polices de texte

Pour les polices de texte de Kp-fonts les options suivantes sont proposées :

**Numbers=Lowercase** : pour avoir les chiffres minuscules ou elzéviens 1,2,3 ou **Numbers=Lining** (par défaut 1,2,3).

**SmallCapsFeatures = {Letters=SmallCaps}** : pour que la commande `\textsc{}` affiche les GRANDES petites capitales à la place des PETITES obtenues par défaut (option **SmallCapsFeatures = {Letters=PetiteCaps}** <sup>4</sup>).

**Ligatures=TeX** : (par défaut) !` ?` -- --- affichent respectivement ¡ ¿ – et —.

**Ligatures=Common** : (par défaut) ligatures automatiques ff ffi ffl fi fl.

**StylisticSet=1** modifie l'aspect des ligatures ffi ffl fi fl (ff inchangé).

**Ligatures=Required** : ajoute les ligatures ft et tt.

**Ligatures=Rare** : ajoute les ligatures ct et st.

**Style=Swash** : pour le Q long également en petites et grandes capitales QUE et QUE.

**Style=Historic** : pour remplacer systématiquement le « s » rond par un « f » long, il reste possible de forcer un « s » rond, en fin de mot notamment, en codant `s=`; ce n'est pas toujours nécessaire sous LuaTeX, voir l'option `veryoldstyle` p. 3.

**Ligatures=Historic** : active les ligatures spécifiques au « f » long fi, fl, ft.

**StylisticSet=2** : `\SS` affiche SS au lieu de  $\beta$  (*eszett* majuscule), idem en petites capitales.

**StylisticSet=3** : remplace le caractère Euro (€) par le symbole officiel (€), disponible également en italique, gras et gras italique.

Le choix parmi ces options se fait normalement globalement pour tout le document, police par police éventuellement, exemple :

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

mais si les options sont communes à plusieurs polices on peut ne les énoncer qu'une fois en procédant de la façon suivante :

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

---

4. Accolades indispensables à cause du signe =!

Remarques : 1. Les commandes `\setmonofont{KpMono}`, `\setsansfont{KpSans}` et `\setmainfont{KpRoman}` fonctionnent parce que la distribution Kp-fonts est fournie avec des fichiers `KpMono.fontspec`, `KpSans.fontspec` et `KpRoman.fontspec`.

2. Noter le + après `\defaultfontfeatures` : les options sont *ajoutées* au options par défaut.

3. On peut regrouper des options : `Ligatures={Rare,Historic}` (avec accolades) équivaut aux deux options `Ligatures=Rare` et `Ligatures=Historic`.

4. Il est possible de modifier une option *localement*, ainsi pour afficher les chiffres elzéviens 1234576890 alors que l'option globale est `Numbers=Lining`, on utilise la commande `\addfontfeatures` dans un groupe :

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

## 2.2 Options des polices mathématiques

Les options suivantes peuvent être passées à `unicode-math`<sup>5</sup> ou à la commande `\setmathfont{}` :

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

`bold-style = ISO, TeX` (défaut), `upright` ;

`partial = upright` (italique par défaut) ;

`nabla = italic` (droit par défaut) ;

Les variantes de style affectent chacune une famille de glyphes, voici les trois premières :

`Style=mathcal (+ss01)` les commandes `\mathcal{}` et `\mathscr{}` produisent  $ABC$  au lieu de  $\mathcal{ABC}$  ;

`Style=bbsans (+ss02)` les glyphes produits par la commande `\mathbb{}` sont en sans-serif  $ABC$  au lieu de  $\mathbb{ABC}$  ;

`Style=narrowiints (+ss03)` les intégrales multiples sont plus compactes :  $\iiint$  au lieu de  $\iiint$ .

---

5. Voir la documentation [unicode-math.pdf](#).

Les quatre tables suivantes recensent les autres variantes de style proposées.

TABLE 1 – Style=leqslant (+ss04)

Commande	Base	Variante
<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>\leqless</code>	$\leqless$	$\leqless$
<code>\eqgtr</code>	$\eqgtr$	$\eqgtr$
<code>\lesseqgtr</code>	$\lesseqgtr$	$\lesseqgtr$
<code>\gtreqless</code>	$\gtreqless$	$\gtreqless$
<code>\lesseqqgtr</code>	$\lesseqqgtr$	$\lesseqqgtr$
<code>\gtreqqlless</code>	$\gtreqqlless$	$\gtreqqlless$

TABLE 2 – Style=smaller (+ss05)

Commande	Base	Variante
<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)

Commande	Base	Variante
<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)

Commande	Base	Variante
<code>\parallel</code>	$\parallel$	$\parallel$
<code>\nparallel</code>	$\nparallel$	$\nparallel$
<code>\shortparallel</code>	$\parallel$	$\parallel$
<code>\nshortparallel</code>	$\nparallel$	$\nparallel$

Exemple : pour activer les styles 4 (leqslant) et 6 (subsetneq) on code `\setmathfont{KpMath-Regular.otf}[StylisticSet={4,6}]` ou bien `\setmathfont{KpMath-Regular.otf}[Style={leqslant,subsetneq}]` mais cette variante plus parlante n'est disponible que si `kpfonts-otf.sty` a été chargé (éventuellement avec l'option `symbols`).

La table 5 page suivante recense les variantes de glyphes proposées.

Exemple : `\setmathfont{KpMath-Regular.otf}[CharacterVariant={3,6}]` fera en sorte que les commandes `\epsilon` et `\phi` produisent  $\epsilon$  et  $\phi$  plutôt que  $\epsilon$  et  $\phi$ . Ceci vaut pour toutes les formes et graisses (droit, italique, gras, gras italique, sans-serif, etc.), ainsi avec l'option `math-syle=french`, `\epsilon` et `\phi` produisent  $\epsilon$  and  $\phi$ .

Note sur `\hbar` : pour `unicode-math` les commandes `\hbar` et `\hslash` affichent le même caractère Unicode U+210F tandis que `amsmath` choisit deux définitions différentes pour ces commandes (un h italique avec une barre horizontale ou oblique). `kpfonts-otf` suit `unicode-math`, le h italique avec barre horizontale peut être obtenu grâce à la variante `cv01` (code `\hslash` ou `\hbar`) ou directement en codant `\mit\hbar`.

TABLE 5 – Variantes de glyphes

	Base		Variante		Commande	
cv00	$\Re$	$\Im$	$\Re$	$\Im$	<code>\Re</code>	<code>\Im</code>
cv01	$\hbar$		$\hbar$		<code>\hslash</code> ou <code>\hbar</code>	
cv02	$\emptyset$		$\emptyset$		<code>\emptyset</code>	
cv03	$\epsilon$		$\epsilon$		<code>\epsilon</code>	
cv04	$\kappa$		$\kappa$		<code>\kappa</code>	
cv05	$\pi$		$\pi$		<code>\pi</code>	
cv06	$\phi$		$\phi$		<code>\phi</code>	
cv07	$\rho$		$\rho$		<code>\rho</code>	
cv08	$\sigma$		$\sigma$		<code>\sigma</code>	
cv09	$\theta$		$\theta$		<code>\theta</code>	
cv10	$\Theta$		$\Theta$		<code>\Theta</code>	

### 3 Commandes spécifiques à Kp-fonts

#### 3.1 Variantes des symboles d'intégration

Kp-fonts propose des variantes des signes intégrales spécialement adaptées à la présentation des primitives, elles se codent `\varint`, `\variint`, `\variiint`, `\variiiiint` et `\varidotsint`. Comparer  $\int f(t) dt$  et  $\int f(t) dt$  et aussi

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

On peut les coupler à la commande `\D{...}`, qui affiche un « d » droit et optimise les espacements : `\displaystyle\varint f(t)\D{t}` donne  $\int f(t) dt$ .

#### 3.2 Symboles spécifiques

Rappelons que la liste des symboles Unicode présents dans Kp-fonts est disponible dans le fichier `unimath-kpfonts.pdf`.

Sauf erreur de ma part les symboles présentés dans les tableaux suivants, n'ont pas de code Unicode, ils sont codés en zone privée pour Kp-fonts.

<code>\mmapsto</code>	$\mapsto$	<code>\longmmapsto</code>	$\longmapsto$
<code>\mmapsfrom</code>	$\mapsfrom$	<code>\longmmapsfrom</code>	$\longmapsfrom$
<code>\Mmapsto</code>	$\Mmapsto$	<code>\Longmmapsto</code>	$\Longmmapsto$
<code>\Mmapsfrom</code>	$\Mmapsfrom$	<code>\Longmmapsfrom</code>	$\Longmapsfrom$
<code>\leftrightdasharrow</code>	$\leftrightarrow$	<code>\leadsto</code>	$\leadsto$



<code>\boxright</code>	$\square\rightarrow$	<code>\boxleft</code>	$\leftarrow\square$
<code>\circleright</code>	$\circ\rightarrow$	<code>\circleleft</code>	$\leftarrow\circ$
<code>\Diamondright</code>	$\diamond\rightarrow$	<code>\Diamondleft</code>	$\leftarrow\diamond$
<code>\boxdotright</code>	$\square\rightarrow\cdot$	<code>\boxdotleft</code>	$\leftarrow\square\cdot$
<code>\circledotright</code>	$\circ\rightarrow\cdot$	<code>\circledotleft</code>	$\leftarrow\circ\cdot$
<code>\Diamonddotright</code>	$\diamond\rightarrow\cdot$	<code>\Diamonddotleft</code>	$\leftarrow\diamond\cdot$

<code>\boxRight</code>	$\square\Rightarrow$	<code>\boxLeft</code>	$\Leftarrow\square$
<code>\boxdotRight</code>	$\square\Rightarrow\cdot$	<code>\boxdotLeft</code>	$\Leftarrow\square\cdot$
<code>\DiamondRight</code>	$\diamond\Rightarrow$	<code>\DiamondLeft</code>	$\Leftarrow\diamond$
<code>\DiamonddotRight</code>	$\diamond\Rightarrow\cdot$	<code>\DiamonddotLeft</code>	$\Leftarrow\diamond\cdot$

<code>\multimapdot</code>	$\multimap\cdot$	<code>\multimapdotinv</code>	$\cdot\multimap$
<code>\multimapdotboth</code>	$\multimap\cdot\cdot$		
<code>\multimapbothvert</code>	$\begin{array}{c} \circ \\ \vdots \\ \circ \end{array}$	<code>\multimapdotbothvert</code>	$\begin{array}{c} \cdot \\ \vdots \\ \cdot \end{array}$
<code>\multimapdotbothAvert</code>	$\begin{array}{c} \circ \\ \vdots \\ \circ \end{array}$	<code>\multimapdotbothBvert</code>	$\begin{array}{c} \cdot \\ \vdots \\ \cdot \end{array}$

<code>\capplus</code>	$\cap+$	<code>\sqcupplus</code>	$\sqcup+$	<code>\sqcappplus</code>	$\sqcap+$
<code>\parallelslant</code>	$\parallel$	<code>\colonsim</code>	$\sim$	<code>\colonapprox</code>	$\approx$
<code>\parallelbackslant</code>	$\backslash\parallel$	<code>\Colonsim</code>	$\sim$	<code>\Colonapprox</code>	$\approx$
<code>\eqqColon</code>	$\equiv$	<code>\Colondash</code>	$\dashv$	<code>\dashColon</code>	$\dashv$

<code>\strictif</code>	$\rightarrow$	<code>\strictfi</code>	$\leftarrow$	<code>\strictiff</code>	$\leftrightarrow$
<code>\circledvee</code>	$\circ\vee$	<code>\circledwedge</code>	$\circ\wedge$	<code>\circledbar</code>	$\circ\bar{\vee}$
<code>\openJoin</code>	$\times$	<code>\opentimes</code>	$\times$	<code>\VvDash</code>	$\Vdash$
<code>\lambdaslash</code>	$\lambda$	<code>\lambdabar</code>	$\bar{\lambda}$	<code>\Wr</code>	$\rightsquigarrow$

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

### 3.3 Accents extensibles

— `\widehat` et `\widetilde`

$\hat{x}$   $\hat{xx}$   $\hat{xxx}$   $\hat{xxxx}$   $\hat{xxxxx}$   $\hat{xxxxxx}$   $\tilde{x}$   $\tilde{xx}$   $\tilde{xx}$   $\tilde{xxx}$   $\tilde{xxxx}$   $\tilde{xxxxx}$

— `\overline` et `\underline`

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

— `\wideoverbar`, `\widecheck` et `\widebreve`

$\bar{x}$   $\bar{xy}$   $\bar{xyz}$   $\check{x}$   $\check{xxx}$   $\check{xxxx}$   $\check{x}$   $\check{xxx}$   $\check{xxxx}$

— `\overparen` et `\underparen`

$\bar{x}$   $\bar{xy}$   $\bar{xyz}$   $\overset{\circ}{A \cup B}$   $\overset{\circ}{A \cup (B \cap C) \cup D}$   $\overset{2}{x+y}$   $\overset{26}{a+b+\dots+z}$

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

— `\overbrace` et `\underbrace`

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

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

— `\overrightarrow` et `\overleftarrow`

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

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

— Enfin `\widearc` et `\widearcarrow` (ou `\overrightarrowarc`)

$\widearc{AMB}$   $\widearcarrow{AMB}$

### 3.4 « Versions » math

Il est possible de définir plusieurs versions de la police mathématique **KpMath**, les plus courantes étant une version grasse (*bold*) et une version sans empattements (*sans*).

On définit dans le préambule les versions désirées, par exemple :

`\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 ]`  
 et ensuite on passe d'une version à l'autre grâce à la commande `\mathversion{nom}`.

Exemple<sup>6</sup> :

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

Voici la même équation en trois versions, normale, grasse et sans empattement :

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

`\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}`

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

---

6. L'option `CharacterVariant=3` remplace le  $\epsilon$  par  $\varepsilon$ .