

# Kp-Fonts

Le projet JOHANNES KEPLER

CHRISTOPHE CAIGNAERT  
c.caignaert@free.fr

## 1. Kp-Fonts

### 1.1. Présentation

Kp-Fonts est un ensemble le plus complet possible de polices pour LaTeX, incluant des polices de texte romaines, sans-serif et machine à écrire, ainsi qu'un ensemble de polices mathématiques comprenant, en normal et gras, tous les symboles usuels et tous les symboles de l'AMS et davantage...

La première caractéristique recherchée de ces polices était d'avoir un design à la fois basique et dynamique. Regardons par exemple le « a » romain :

Kp-Fonts	CM	Palatino	Utopia	Times
a	a	a	a	a

Les autres polices ont été dessinées de façon à s'harmoniser au mieux. Kp-Fonts ne nécessite a priori aucun autre package de polices et est complètement compatible avec `amsmath` que Kp-fonts charge ainsi que `textcomp`.

*Vous ne devez donc pas charger les extensions `amsmath` et `textcomp`.*

L'utilisation de Kp-Fonts ne nécessite qu'un `\usepackage{kpfonts}` dans le préambule de votre document, au besoin avec des options décrites plus loin. Enfin, on a recherché une compatibilité maximale avec différents ensembles de polices mathématiques existants.

### 1.2. Polices de texte

- Les codages `τ1`, `ot1` et `ts1` sont disponibles complètement à une petite exception près pour le symbole `\SS` en majuscule dans les polices de type machine à écrire ;
- Les polices romaines sont disponibles droites, en petites majuscules, italiques et penchées en normal, gras et gras étendu ;
- Les polices sans-serif sont disponibles droites, en petites majuscules et penchées en normal, gras et gras étendu ;
- Les polices télétypes sont disponibles droites et penchées, en normal et gras.

*La graisse par défaut est **bold**, et non pas **bold extended**.*

### 1.3. Polices mathématiques

- La commande `\boldmath` est complètement implantée et Kp-Fonts **ne nécessite pas** la commande `\bm` ;
- On dispose de polices grecques droites  $\alpha\beta\gamma\Gamma\Delta$  et penchées  $\alpha\beta\gamma\Gamma\Delta$  ;
- La commande `\mathbb` fournit  $\mathbb{A}\mathbb{B}\mathbb{C}$ , caractères dessinés à partir des majuscules de texte ;
- La commande `\mathcal` fournit  $\mathcal{A}\mathcal{B}\mathcal{C}$ , version légèrement modifiée des polices CM ;
- La commande `\mathscr` fournit  $\mathscr{A}\mathscr{B}\mathscr{C}$ , sans aucun package supplémentaire ; ces caractères ont été dessinés à partir de polices traditionnelles ;
- La commande `\mathfrak` fournit  $\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{a}\mathfrak{b}\mathfrak{c}$ , toujours à partir de polices traditionnelles redessinées pour être plus lisibles.

## 2. Options

### 2.1. Options globales

**nomath** : Kp-Fonts ne modifie pas les polices les polices mathématiques par défaut ;

**notext** : Kp-Fonts ne modifie pas les polices de texte par défaut.

Et si vous utilisez ces deux options, Kp-Fonts ne fait rien...

### 2.2. Options des polices de texte

**nosf** : Kp-Fonts ne modifie pas les polices sans-serif par défaut ;

**nott** : Kp-Fonts ne modifie pas les polices machine à écrire par défaut.

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

### 2.3. Lettres grecques en mode mathématique

**uprightgreek** : les *minuscules* grecques sont droites,  $\alpha\beta\gamma$  ;

**slantedGreek** : les *Majuscules* grecques sont penchées,  $\Gamma\Delta$ .

### 2.4. Options des polices mathématiques

**nomathscript** : Kp-Fonts ne crée pas la commande `\mathscr` ;

**mathcalasscript** : cette option échange les polices `\mathcal` et `\mathscr` ;

**classicReIm** : les commandes `\Re` et `\Im` donnent comme habituellement  $\Re$  et  $\Im$ . Par défaut, Kp-Fonts donne  $\mathbb{R}$  et  $\mathbb{I}$  ;

**uprightRoman** : les *Majuscules* romaines sont droites.

**frenchstyle** : équivaut aux options **uprightRoman** et **uprightgreek**, les majuscules romaines et les minuscules grecques sont droites, selon la tradition typographique française ; les minuscules romaines restent obliques ;

**upright** : synonyme de la précédente.

### 3. Utilisation

#### 3.1. Accès aux caractères grecs

On a vu les deux options qui permettent de modifier les caractères grecs utilisés par défaut.

Par ailleurs, on peut toujours accéder aux autres caractères grecs par les commandes `\otheralpha`, `\otherGamma`. Le résultat dépend ici des options utilisées.

Pour les majuscules, on dispose aussi des commandes du type `\varGamma` qui sont synonymes des précédentes.

Enfin, les commandes comme `\alpha1`, `\alphaup`, `\Gammaup` ou `\Gamma1` permettent toujours d'obtenir le caractère désiré quelle que soit l'option utilisée.

#### 3.2. Symboles standards

Tous les symboles standards sont présents, y compris tous ceux du package `amssymb`, en graisse normale et grasse.

Par exemple :  $\leq$   $\bullet$   $\pm$  pour les symboles standards ou  $\Rightarrow$   $\subsetneq$   $\leqslant$  pour les symboles de l'AMS.

et aussi en gras :  $\leq$   $\bullet$   $\pm$  ou  $\Rightarrow$   $\subsetneq$   $\leqslant$ .

On a bien sûr aussi les constructions habituelles en normal ou gras :  $\mapsto$ ,  $\mapsto$ ,  $\mapsto$ .

#### 3.3. Symboles supplémentaires

Kp-Fonts propose de nombreux symboles ou constructions supplémentaires.

On a aussi de nombreuses négations non reportées ici.

<code>\mappedfrom</code>	$\leftarrow$	<code>\longmappedfrom</code>	$\longleftarrow$
<code>\Mapsto</code>	$\Rightarrow$	<code>\Longmapsto</code>	$\Longrightarrow$
<code>\Mappedfrom</code>	$\Leftarrow$	<code>\Longmappedfrom</code>	$\Longleftarrow$
<code>\mmapsto</code>	$\Rrightarrow$	<code>\longmmapsto</code>	$\LongRrightarrow$
<code>\mmappedfrom</code>	$\Leftarrow$	<code>\longmmappedfrom</code>	$\Longleftarrow$
<code>\Mmapsto</code>	$\Rightarrow$	<code>\Longmmapsto</code>	$\Longrightarrow$
<code>\Mmappedfrom</code>	$\Leftarrow$	<code>\Longmmappedfrom</code>	$\Longleftarrow$
<code>\dashleftarrow</code>	$\dashleftarrow$	<code>\dashrightarrow</code>	$\dashrightarrow$
<code>\dashleftrightharpoonup</code>	$\leftrightarrow$	<code>\leftsquigarrow</code>	$\leftrightsquigarrow$
<code>\Nearrow</code>	$\nearrow$	<code>\Searrow</code>	$\searrow$
<code>\Nwarrow</code>	$\nwarrow$	<code>\Swarrow</code>	$\swarrow$
<code>\leadstoext</code>	$\leadsto$	<code>\leadsto</code>	$\leadsto$

On notera que ces deux derniers symboles se combinent :

`\leadstoext\leadstoext\leadstoext\leadsto` donne  $\leadsto\leadsto\leadsto$

<code>\boxright</code>	$\square\rightarrow$	<code>\Diamondright</code>	$\diamondrightarrow$
<code>\circcleright</code>	$\circ\rightarrow$	<code>\boxleft</code>	$\leftarrow\square$
<code>\Diamondleft</code>	$\leftarrow\diamond$	<code>\circleleft</code>	$\leftarrow\circ$
<code>\boxdotright</code>	$\boxdot\rightarrow$	<code>\Diamonddotright</code>	$\diamondrightarrow$
<code>\circleddotright</code>	$\circ\rightarrow$	<code>\boxdotleft</code>	$\leftarrow\boxdot$
<code>\Diamonddotleft</code>	$\leftarrow\diamond$	<code>\circleddotleft</code>	$\leftarrow\circ$
<code>\boxRight</code>	$\boxdot\rightarrow$	<code>\boxLeft</code>	$\leftarrow\boxdot$
<code>\boxdotRight</code>	$\boxdot\rightarrow$	<code>\boxdotLeft</code>	$\leftarrow\boxdot$
<code>\DiamondRight</code>	$\diamondrightarrow$	<code>\DiamondLeft</code>	$\leftarrow\diamond$
<code>\DiamonddotRight</code>	$\diamondrightarrow$	<code>\DiamonddotLeft</code>	$\leftarrow\diamond$
<code>\multimap</code>	$\multimap$	<code>\multimapinv</code>	$\multimap$
<code>\multimapboth</code>	$\multimap$	<code>\multimapdot</code>	$\multimap$
<code>\multimapdotinv</code>	$\multimap$	<code>\multimapdotboth</code>	$\multimap$
<code>\multimapdotbothA</code>	$\multimap$	<code>\multimapdotbothB</code>	$\multimap$
<code>\multimapbothvert</code>	$\multimap$	<code>\multimapdotbothvert</code>	$\multimap$
<code>\multimapdotbothAvert</code>	$\multimap$	<code>\multimapdotbothBvert</code>	$\multimap$
<code>\Wr</code>	$\wr$	<code>\sqcupplus</code>	$\sqcupplus$
<code>\medcirc</code>	$\bigcirc$	<code>\medbullet</code>	$\bullet$
<code>\VDash</code>	$\Vdash$	<code>\VvDash</code>	$\Vdash$
<code>\preceq</code>	$\preceq$	<code>\succeq</code>	$\succeq$
<code>\varparallel</code>	$\parallel$	<code>\varparallelinv</code>	$\parallel$
<code>\colonsim</code>	$\sim$	<code>\Colonapprox</code>	$\approx$
<code>\eqqcolon</code>	$\equiv$	<code>\coloneq</code>	$\equiv$
<code>\Coloneqq</code>	$\equiv$	<code>\Eqqcolon</code>	$\equiv$
<code>\Diamonddot</code>	$\diamond$	<code>\Diamond</code>	$\diamond$
<code>\strictif</code>	$\rightarrow$	<code>\strictfi</code>	$\rightarrow$
<code>\circledless</code>	$\circ$	<code>\circledgtr</code>	$\circ$
<code>\circledvee</code>	$\circ$	<code>\circledbar</code>	$\circ$
<code>\lJoin</code>	$\Join$	<code>\rJoin</code>	$\Join$
<code>\openJoin</code>	$\Join$	<code>\lRtimes</code>	$\Join$
<code>\Lbag</code>	$\{$	<code>\Rbag</code>	$\}$
<code>\Top</code>	$\top$	<code>\Bot</code>	$\bot$
<code>\boxast</code>	$\boxtimes$	<code>\boxbslash</code>	$\boxtimes$
<code>\boxslash</code>	$\boxtimes$	<code>\lambdaslash</code>	$\lambda$
<code>\varclubsuit</code>	$\clubsuit$	<code>\vardiamondsuit</code>	$\diamondsuit$
<code>\varspadesuit</code>	$\spadesuit$	<code>\llbracket</code>	$\llbracket$
		<code>\rrbracket</code>	$\rrbracket$

`\lbag`, `\rbag`, `\llbracket` et `\rrbracket` sont verticalement extensibles.

### 3.4. Variantes des symboles d'intégration

On peut remarquer que, quand on écrit une primitive, le résultat manque souvent d'élégance car la fonction est souvent placée trop loin du signe d'intégration.

Ce qu'on peut voir ici :

$$\int f(t)dt$$

Kp-Fonts possède des instructions supplémentaires pour éviter cela. La première est la commande `\varint` qui donne :

$$\int f(t)dt$$

Chacun peut ainsi choisir la forme qui lui convient le mieux !

Bien sûr, cette commande ne convient pas quand on calcule une intégrale...

Cette commande fonctionne aussi en mode texte.

Les autres commandes disponibles sont : `\variint`, `\variiint`, `\variiiiint` et `\varidotsint`.

### 3.5. Grands symboles supplémentaires

Notons d'abord que les commandes `\widehat` et `\widetilde` ont été étendues :



































$$\widetilde{tilde} \quad \widehat{chapeau}$$

Notons les quelques commandes supplémentaires :

<code>\bignplus</code>	⊕	$\bigoplus$	<code>\bigsqcupplus</code>	⊞	$\bigoplus$
<code>\bigsqcapplus</code>	⊞	$\bigoplus$	<code>\bigsqcap</code>	⊞	$\bigoplus$
<code>\varprod</code>	×	$\prod$			

### 3.6. Symboles d'intégration supplémentaires

Il existe de nombreux signes d'intégration supplémentaires :

$\backslash oiint$			$\backslash ointctrackwise$		
$\backslash ointclockwise$			$\backslash sqint$		
$\backslash idotsint$			$\backslash oiiiint$		
$\backslash varointctrackwise$			$\backslash varointclockwise$		
$\backslash fint$			$\backslash oiintctrackwise$		
$\backslash varoiintclockwise$			$\backslash oiintclockwise$		
$\backslash varoiintctrackwise$			$\backslash oiiiintctrackwise$		
$\backslash varoiiiintctrackwise$			$\backslash sqiint$		
$\backslash sqiiiint$					

## 4. Installation

Le dossier `kpfonts` fourni respecte l'arborescence TDS standard. Vous devez d'abord recopier tous ces fichiers dans une de vos arborescences `texmf` en accord avec votre organisation de fichiers puis mettre à jour votre base de données (`mktexlsr` ou, avec MikTeX, par le menu MikTeX-Options).

Il faut maintenant s'occuper du fichier `.map`. Si vous avez une distribution récente respectant les spécifications web2c, il vous suffit d'exécuter `udpmap`.

```
udpmap --enable Map=kpfonts.map
```

Si ceci ne marche pas, il suffit chercher les fichiers `psfonts.map` et `pdfTeX.map`. Ensuite, ajouter le contenu de `kpfonts.map` dans chacun d'entre eux et sauvegarder les ... dans le même répertoire !

Sous MikTeX, le plus simple de faire la même chose avec le ou les fichiers : `texmf/pdftex/config/psfonts.map`, ou, `texmf/dvips/config/psfonts.map` selon ce qu'on utilise.

## 5. Quelques petits compléments

### 5.1. Affichage et impression

Comme souvent, il peut exister des problèmes d'affichage ou d'impression... Les problèmes d'affichage sont dus à un « hinting » automatique. Je ne suis pas typographe professionnel et je ne sais pas intervenir là dessus ; les problèmes d'impression peuvent se poser avec des versions anciennes d'*Acroread*® sous *Windows*®.

Signalons que l'impression est meilleure sous *Ghostscript-Ghostview*®...

## 5.2. Évolution prévue

Il est prévu une option `oldstyle` avec les vieilles ligatures française « ct » et « st ». Quand j'aurai le temps...

Par ailleurs, si vous avez besoin d'un symbole supplémentaire, n'hésitez pas à me contacter, je ferai de mon mieux.

## 5.3. JOHANNES KEPLER 1571-1630

Vers 1600, KEPLER est obligé de fuir à Prague suite aux persécutions religieuses de la contre-réforme, en effet il était luthérien. Il rejoint l'observatoire de l'astronome danois TYCHO BRAHE. Mais celui-ci meurt en 1601, lui laissant l'usage de ses archives. En analysant méthodiquement, et sans a-priori, ces données, il établit les deux premières lois du mouvement planétaire qui ont immortalisé son nom (1609); lois qu'il vérifia par d'autres observations. Il fut ainsi un précurseur de la méthode scientifique.

## 5.4. Dernière remarque

Notons que *Kepler*® est un nom déposé de police distribué par Adobe™. Les polices de Kp-Fonts n'ont rien à voir avec celles-ci.

Kp-Fonts est distribuée sous licence GPL.

## 5.5. Remerciements

Je tiens à remercier pour leur aide et leur soutien

- NICOLAS BOULENGUEZ (Tests);
- MICHEL BOVANI (Polices);
- DANIEL FLIPO (Latex);
- SOURAYA MUHIDINE (Traduction)
- PETER ROSENFELD (URW)