

# Kp-Fonts

Le projet JOHANNES KEPLER

Version 2.3

CHRISTOPHE CAIGNAERT  
c.caignaert@free.fr

La version 2.3 fournit

- une nouvelle option *nofligatures* sans les ligatures *fi*, *fl*, *ff*, *fi* et *ffl* ; les glyphes des *f* ont été redessinées pour obtenir un bon résultat ; il est alors possible de rechercher dans tous les visualisateurs des mots comme *difficile* ;
- des petites majuscules penchées avec la forme *scsl*.

## 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 droit :

Kp-Fonts	Kp-light	CM	Palatino	Utopia	Times

Le projet est de fournir un grand ensemble d'options pour individualiser votre composition de texte ou de mathématiques.

Ce package LaTeX est distribué sous licence GPL.

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` (avec l'option `full`). À moins bien sûr que vous utilisiez les options relatives à ces chargements...

*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`, `or1` et `rs1` sont disponibles complètement à une petite exception près pour le symbole `\SS` en majuscule dans les polices de type machine à écrire ; si vous utilisez des accents, vous devez travailler en `τ1`, le codage `or1` n'est là que pour des raisons historiques ;
- Les polices romaines sont disponibles droites, en petites majuscules, italiques, penchées, petites majuscules penchées et « `oldstyle` » en normal, gras et gras étendu ;
- Les polices sans-serif sont disponibles droites et penchées, en petites majuscules droites et penchées, en « `oldstyle` », 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** de « poor man bold » ;
- 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 :  $\mathfrak{C}\mathfrak{T}\mathfrak{A}\mathfrak{Q}$  à comparer avec CTAN sur [www.ctan.org](http://www.ctan.org) !

# 2. Options

## 2.1. Principales options globales

**light** : alors, vous utilisez 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 ;

**fulloldstylenums** : équivaut aux deux options `oldstylenums` et `oldstylenumsmath` ;

**fulloldstyle** : équivaut aux deux options `oldstyle` et `oldstylemath` ;

**fullveryoldstyle** : équivaut aux deux options `veryoldstyle` et `veryoldstylemath`.

Ce texte est composé avec les polices par défaut, et ci-dessous, vous trouverez un exemple des polices « *light* », en droit et italique, normal et gras :

While the high-level font commands are intended for use in a document, the low-level commands...

While the high-level font commands are intended for use in a document, the low-level commands...

*While the high-level font commands are intended for use in a document, the low-level commands...*

*While the high-level font commands are intended for use in a document, the low-level commands...*

## 2.2. Autres 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 ;

**nosf** : Kp-Fonts ne modifie pas les polices sans-serif par défaut (text et math) ;

**nott** : Kp-Fonts ne modifie pas les polices machine à écrire par défaut (text et math) ;

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

**noamsmath** : Kp-Fonts ne charge pas *amsmath* ;

**notextcomp** : Kp-Fonts ne charge pas *textcomp*.

Et si vous utilisez les deux premières et les deux dernières options, Kp-Fonts ne fait rien...

## 2.3. Options des polices de texte

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

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

Une commande `\classicstylenums` permet alors, en romain, d'accéder aux nombres usuels.

Quelques exemples en droit et italique, medium et gras :

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

**oldstyle** : Les polices fournissent ici « Q », les nombres elzéviens par défaut.

En romain et sans-serif, `oldstyle` fournit les anciennes ligatures « ct » et « st », et enfin, la petite capitale « Q ».

Une commande `\classicstylenums` permet alors, en romain, d'accéder aux nombres usuels.

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` sauf que le « s » rond est remplacé par le « f » long.

On peut toujours accéder au « s » rond par la ligature « s= », ce qui est utilisé assez souvent en fin de mot.

Exemple : `\textit{espoirs=}` donne « *espoirs* »

Evidemment, il n'y pas de « f » long en petites capitales !

**rmx** alors, vous avez six graisses, données ici avec les correspondances :

light	<b>l</b>	medium en mode light
medium	<b>m</b>	medium en mode standard
semi-bold	<b>sb</b>	<b>bold en mode light</b>
bold	<b>b</b>	<b>bold en mode standard</b>
semi-bold extended	<b>sbx</b>	<b>bold extended en mode light</b>
bold extended	<b>bx</b>	<b>bold extended en mode standard</b>

Dans ce cas, l'option *light* n'affecte que les polices mathématiques.

Je n'aime pas trop cette option car les familles standard et light sont cousines et non pas sœurs ! Par ailleurs, cette option n'affecte que les polices romaines.

**largesmallcaps** donne des petites majuscules plus grandes que par défaut :

GRANDES et PETITES (par défaut) petites majuscules !

**nofligatures** donne un texte ne contenant pas les ligatures ff, fi, fl, ffi et ffl.

La lettre *f* a été redessinée pour donner un résultat convenable.

Cette option est ignorée en présence de *oldstyle* ou *veryoldstyle*.

On obtient par exemple, « Un finaliste efflanqué » au lieu d'avoir, « Un finaliste efflanqué ».

De la même façon, « *Un finaliste efflanqué* » au lieu d'avoir, « *Un finaliste efflanqué* ».

## 2.4. Nouvelles commandes de texte

**\textscsl{...}** et **\scslshape** : fournissent les petites majuscules penchées.

*PETITES CAPITALES PENCHÉES*

*Attention, \textsc{\textsl{...}} ne fournit pas ce résultat !*

*Cette forme de police n'est pas habituelle en T<sub>E</sub>X !*

**\otherscshape**, **\textothersc{...}** fournissent les *autres* petites majuscules (normales ou grandes) romaines ou sans-serif.

On échange par exemple

EXEMPLE EN PETITES CAPITALES *et* EXEMPLE EN PETITES CAPITALES

**\otherscslshape**, **\textotherscsl{...}** fournissent les *autres* petites majuscules penchées (normales ou grandes) romaines ou sans-serif.

On échange par exemple

EXEMPLE EN PETITES CAPITALES *et* EXEMPLE EN PETITES CAPITALES

**\othertailQ** change le dessin de la majuscule « Q », avec une courte ou longue queue.

On échange par exemple

Question et Question

`\othertailscq`, `\othertailscslq` font la même chose avec les petites capitales « Q ».

On échange par exemple

QUESTION et QUESTION

*Les commandes « other » changent la **taille** des petites capitales  
et les commandes « othertail » changent le **dessin** des lettres « Q »*

## 2.5. Options des polices mathématiques

**nomathscript** : Kp-Fonts ne crée pas la commande `\mathscr` ; cette option vous est utile si vous utiliser le package `\mathrsfs` ;

**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  $\Re$  et  $\Im$  ;

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

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

**upright** : synonyme de la précédente ;

**oldstylenumsmath** : fournit les nombres elzéviens par défaut et change les polices de `\mathrm`, `\mathsf` et `\mathtt` qui fonctionnent comme les polices de texte avec l'option `oldstylenums` ;

**oldstylemath** : fournit les nombres elzéviens par défaut et change les polices de `\mathrm`, `\mathsf` et `\mathtt` qui fonctionnent comme les polices de texte avec l'option `oldstyle` ;

**veryoldstylemath** : identique à **oldstylemath** avec, en plus, le « s » long ;

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

$\int\int\int$  et  $\int\int\int$  au lieu de  $\int\int\int$  et  $\int\int\int$

**partialup** : fournit le symbole `\partial` droit :

$\partial$  au lieu de  $\partial$

**noDcommand** : pour préserver une éventuelle compatibilité avec un autre package, la commande `\D` n'est pas chargée.

Notons que, avec les options **oldstylenumsmath**, **oldstylemath**, **veryoldstylemath**, **fulloldstylenums**, **fulloldstyle** ou **fullveryoldstyle**, en mode mathématique, comme les nombres 3, 4, 5, 7 et 9 sont descendants, les exposants sont placés plus hauts.

## 2.6. Options de gestion des lettres grecques en mode mathématique

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

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

### 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 grasse 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>\Mappedfrom</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>\vareemptyset</code>	$\emptyset$		
<code>\leadstoext</code>	$\leadsto$	<code>\leadsto</code>	$\leadsto$

On notera que ces deux derniers symboles se combinent :

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

<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>	$\square\rightarrow$	<code>\Diamonddotright</code>	$\diamondrightarrow$
<code>\circleddotright</code>	$\circ\rightarrow$	<code>\boxdotleft</code>	$\leftarrow\square$
<code>\Diamonddotleft</code>	$\leftarrow\diamond$	<code>\circleddotleft</code>	$\leftarrow\circ$
<code>\boxRight</code>	$\square\Rightarrow$	<code>\boxLeft</code>	$\Leftarrow\square$
<code>\boxdotRight</code>	$\square\Rightarrow$	<code>\boxdotLeft</code>	$\Leftarrow\square$
<code>\DiamondRight</code>	$\diamond\Rightarrow$	<code>\DiamondLeft</code>	$\Leftarrow\diamond$
<code>\DiamonddotRight</code>	$\diamond\Rightarrow$	<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>	$\colon\sim$	<code>\Colonapprox</code>	$\colon\approx$
<code>\eqqcolon</code>	$\equiv$	<code>\coloneq</code>	$\coloneq$
<code>\Coloneqq</code>	$\equiv$	<code>\Eqqcolon</code>	$\equiv$
<code>\Diamonddot</code>	$\diamond$	<code>\Diamond</code>	$\diamond$
<code>\Diamondblack</code>	$\blacklozenge$		
<code>\strictif</code>	$\rightarrowtail$	<code>\strictfi</code>	$\leftarrowtail$
<code>\circledless</code>	$\circledless$	<code>\circledgtr</code>	$\circledgtr$
<code>\circledvee</code>	$\circledvee$	<code>\circledbar</code>	$\circledbar$
<code>\lJoin</code>	$\Join$	<code>\rJoin</code>	$\Join$
<code>\openJoin</code>	$\Join$	<code>\lRtimes</code>	$\Join$
<code>\Lbag</code>	$\Join$	<code>\Rbag</code>	$\Join$
<code>\Top</code>	$\top$	<code>\Bot</code>	$\bot$
<code>\boxast</code>	$\boxast$	<code>\boxbslash</code>	$\boxast$
<code>\boxslash</code>	$\boxast$	<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...

On peut aussi la coupler à la commande `\D{...}`, le « d » de l'intégration avec un espacement optimisé :

$$\text{\displaystyle\varint f(t)\D{t}} \quad \text{donne} \quad \int f(t) dt$$

Avec l'option *frenchstyle*, on obtient un « d » droit, comme plus haut.

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 :

$$\widetilde{t\grave{ilde}} \quad \widehat{chapeau}$$



































Notons les quelques commandes supplémentaires :

<code>\bignplus</code>	⊕	⏞ +		<code>\bigsqcupplus</code>	⊞	⏞ +
<code>\bigsqcappplus</code>	⊞	⏞ +		<code>\bigsqcap</code>	⊞	⏞ ⊞
<code>\varprod</code>	×	⊗				

### 3.6. Symboles d'intégration supplémentaires

Il existe de nombreux signes d'intégration supplémentaires :



<code>\oiint</code>			<code>\ointctrclockwise</code>		
<code>\ointclockwise</code>			<code>\sqint</code>		
<code>\idotsint</code>			<code>\oiint</code>		
<code>\varointctrclockwise</code>			<code>\varointclockwise</code>		
<code>\fint</code>			<code>\oiintctrclockwise</code>		
<code>\varoiintclockwise</code>			<code>\oiintclockwise</code>		
<code>\varoiintctrclockwise</code>			<code>\oiintctrclockwise</code>		
<code>\varoiintctrclockwise</code>			<code>\sqint</code>		
<code>\sqiiint</code>					

## 4. Installation

- Avec MikTeX, installez le package comme décrit ici :

<http://docs.miktex.org/manual/pkgmgt.html#id562117>

- Avec une autre distribution, ou, sous MikTeX, pour faire une installation manuelle, suivez ces instructions :

1. Le package fournit une architecture TDS standard. Vous devez copier les fichiers dans l'une de vos arborescences `texmf` locales puis mettre à jour votre base de fichiers.

2. Maintenant, vous devez mettre à jour vos fichiers `.map`.

- Avec une distribution `web2c`, exécutez `updmap` :

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

et/ou, cette fois ci en tant que *root* :

```
updmap-sys --enable Map=kpfonts.map
```

- Sous MikTeX, suivez les instructions du manuel :

<http://docs.miktex.org/manual/advanced.html#psfonts>

## 5. Quelques petits compléments

### 5.1. Les familles de polices de texte du projet JOHANNES KEPLER

Si vous voulez, ou si vous devez, utiliser les commandes de bas niveau, les noms des familles de polices sont constitués de la façon suivante :

<b>roman</b>	jkp[l,x][k][f][osn,os,vos]
<b>sans serif</b>	jkpss[k][f][osn,os,vos]
<b>télétype</b>	jkptt[osn,os,vos]

où les options correspondantes sont :

<b>l, x</b>	light, rmx
<b>k</b>	largesmallcaps
<b>f</b>	nofligatures
<b>osn, os, vos</b>	oldstylenums, oldstyle, veryoldstyle

Bien sûr, on ne peut avoir **f** et **os** ou **vos**.

Remarquons que toutes ces familles existent dans les encodages OT1, T1 et TS1.

### 5.2. Métriques et compatibilité

- L'option *light* ne change aucune métrique ;
- Les familles d'options *oldstylenums* et *oldstyle* ne changent pas de métrique horizontale ;
- La famille d'options *veryoldstyle* change aussi les métriques du 's' minuscule et la hauteur des exposants en mode math ;
- Toutes les options sont compatibles, sauf si cela n'a pas de sens, comme les deux options *nomath* et *frenchstyle* simultanément ;
- En cas de conflit entre des options des familles *oldstylenums*, *oldstyle* et *veryoldstyle*, les options les plus légères sont ignorées.

### 5.3. 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*®.

Si vous utilisez l'option *light*, l'impression est nettement meilleure que l'affichage !

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

#### 5.4. Résumé

L'ensemble des possibilités de *kpfonts* est résumé en une page dans le fichier *kpfonts-abstract.pdf* !

#### 5.5. Mes options préférées

- Pour un document sans math, j'utilise les options *light* et *oldstyle* ;
- et pour un document avec des maths, j'utilise les options *light*, *frenchstyle*, *narrowiints* et *partialup*, mais je suis aussi français !

#### 5.6. 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.

#### 5.7. 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.

#### 5.8. Remerciements

Je tiens à remercier pour leur aide et leur soutien

- NICOLAS BOULENGUEZ (Tests) ;
- MICHEL BOVANI (Polices) ;
- DANIEL FLIPO (L<sup>A</sup>T<sub>E</sub>X) ;
- SOURAYA MUHIDINE (Relecture de la traduction)
- PETER ROSENBERG (URW)
- CHRISTIAN TELLECHEA (package *xstring*)