

Débuter en



Yves Soulet – Hiver 2016

Partie II.B – Gestion des fontes

Table des matières

B	Gestion des fontes	1
B.1	Éléments de typographie numérique	1
B.2	Fichiers de fontes dans le monde L ^A T _E X	1
B.3	Attributs d'une fonte	3
B.3.1	Attribut codage	3
B.3.2	Attribut famille	4
B.3.3	Attribut graisse-largeur	4
B.3.4	Attribut forme	4
B.3.5	Attribut corps-interligne	5
B.4	Fontes texte et manipulations élémentaires	5
B.4.1	Fontes disponibles par défaut	6
B.4.2	Ajouter des fontes de la famille utilisée	9
B.4.3	Ajouter une deuxième fonte principale	10
B.5	Recherche et installation d'une fonte	10
B.5.1	Symboles supplémentaires	10
B.5.2	Préparation des tests	11
B.5.3	Exemples de fontes disponibles	12
B.6	Installer une nouvelle fonte	16
B.6.1	Fichiers de fontes à créer	17
B.6.2	Programmes utilisés	18
B.6.3	Création des fichiers de fontes	18
	Bibliographie	20
	Index	21

Gestion des fontes

Cette partie est destinée, à plonger dans la « partie fontes » de \LaTeX , c'est-à-dire dans ce qu'on appelle la NFSS mentionné à la fin du chapitre 1, page 4. Avant de commencer il est utile de rappeler ou d'exposer quelques notions sur l'impression.

B.1 Éléments de typographie numérique

Les imprimantes impriment des points (en noir ou en couleur). Il y a les imprimantes à aiguilles où des aiguilles frappent le papier au travers d'un ruban imbibé d'encre (cf. les anciennes machines à écrire). Il y a aussi les imprimantes laser qui déposent sur le papier de minuscules paquets de toner ; ce toner est ensuite « cuit » pour qu'il adhère au papier et forme aussi des points. Dans le deuxième cas, si ces points sont bien placés et assez petits, le lecteur voit de magnifiques lettres bien nettes ; dans le premier cas, on obtient des résultats lisibles dont on s'est contenté pendant des années.

La qualité de l'impression, que l'on nomme la « résolution », s'exprime en nombre de points imprimés par pouce (p/in) ; dans le domaine de l'imprimerie on utilise les unités suivantes :

1 pouce (in) = 72 points (pt) \approx 25 mm ; donc 1 mm \approx 3 pt.

Il faut veiller à ne pas confondre le point imprimé et le point unité de longueur.

La résolution des imprimantes à aiguilles (encore commercialisées et utilisées pour certains travaux) ne dépasse pas quelques dizaines de points par pouce. Par contre la résolution des imprimantes laser est actuellement de 1200 p/in et l'imprimerie professionnelle de haute qualité travaille en 4800 p/in.

B.2 Fichiers de fontes dans le monde \LaTeX

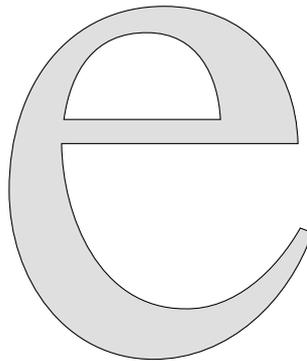
Il y a d'abord les fichiers `.tfm` appelées fichiers métriques `\.tfm` car ils contiennent toutes les dimensions des caractères de la fonte, dimensions nécessaires pour faire la composition (cf. la compilation, section 2.1.2, page 8). Ils contiennent encore les indications pour les crénages, c'est-à-dire les rapprochements de certains couples de caractères (cf. section 4.3, page 25) mais encore bien d'autres dimensions, par exemple la hauteur à laquelle se place l'exposant (dimensions que les \TeX perts peuvent cependant modifier à la demande).

Il y a ensuite les fichiers qui contiennent le dessin des caractères qui seront « intégrés » dans la sortie finale : ces fichiers sont de deux types

- les fichiers du type *bitmap* qui contiennent tout simplement (sous forme compactée) les coordonnées des points à imprimer. Dans le monde L^AT_EX, ce sont les fichiers `.pk` introduit à la section 2.1.6, page 9. On prend l'exemple de la lettre « o » du présent document : sa hauteur et sa largeur sont proches de 2 mm. Pour la résolution 300 p/in, soit 12 points par mm, le créateur de la fonte va tracer une grille de 24 lignes de 24 colonnes, tracer les contours intérieur et extérieur de la lettre et enfin prendre les coordonnées des petits carrés sur lesquels sera imprimé un point. On découvre tout de suite l'important problème de la « rasterisation » : il doit décider quels sont, parmi les petits carrés traversés par les limites, ceux sur lesquels sera imprimé un point et ceux qui resteront blancs.

Il y a bien sûr des utilitaires puissants qui aident à faire ce travail, y compris la rasterisation. Mais il reste encore un problème de taille : il faut un fichier pour chaque résolution et même pour chaque imprimante car l'expérience a montré qu'il faut adapter la rasterisation au type d'imprimante⁽¹⁾. Par exemple, pour une résolution de 4800 p/in, 16 fois plus grande que 300 p/in, on aura des fichiers 256 fois plus gros ... et il ne faut pas oublier que, par principe même, il faut un fichier pour chaque taille et que ces fichiers croissent aussi avec le carré de la taille !

- Heureusement il y a les fontes *vectorielles* qui, pour le monde L^AT_EX, sont des fontes dites de *type 1* dont les fichiers ont les extensions `.pfa` ou `.pfb`. Ces fichiers, déjà annoncés à la section 2.1.6, page 9, ne contiennent que les contours intérieurs et extérieurs des caractères et *c'est le processeur de l'imprimante qui décide des points à imprimer, donc qui assure la rasterisation.*



Les contours sont en général constitués par des courbes de Béziérs⁽²⁾. Ces courbes sont définies par quatre points : origine, extrémité et deux points dits de contrôle. L'exemple proposé ci-dessus⁽³⁾ est constitué par huit courbes dont les points de

⁽¹⁾ Les principes sont toujours les mêmes, mais les réalisations techniques ne sont pas identiques d'une marque à l'autre : il faut donc adapter la rasterisation à chaque imprimante.

⁽²⁾ Béziérs a utilisé des courbes du troisième degré pour dessiner des capots de voiture chez Renault ; ces courbes portent maintenant son nom ; s'il a fait de beaux capots, il a aussi rendu service à la typographie par la promotion involontaire qu'il a faite de ces courbes.

⁽³⁾ Exemple cité par J. André, Cahiers GUTenberg, **26**, p. 24 et présenté ici codé en TikZ).

contrôle entourent les mots `and` dans le code et trois segments repérables par les doubles tirets `--` : cela fait 27 points géométriques seulement. Voici le codage (on notera aussi la magnifique collaboration entre l'informatique et l'art) :

```
\begin{tikzpicture}[x=0.1mm,y=0.1mm]
\draw[fill=gray!25] (402,276)..controls(399,380)and(334,458)
..(226,458)..controls(102,458)and(22,356)..(22,214)..controls
(22,95)and(97,-10)..(212,-10)..controls(312,-10)and(390,68)..
(421,158)--(405,164)..controls(374,109)and(319,55)..(253,57)
..controls(140,57)and(92,181)..(91,276)--cycle;
\draw[fill=white] (94,308)..controls(103,372)and(134,424)..
(204,423)..controls(270,423)and(297,366)..(300,308)--cycle;
\end{tikzpicture}
```

B.3 Attributs d'une fonte

La NFSS définit cinq attributs affectés par les commandes primitives suivantes :

- `\fontencoding{codage}`;
- `\fontfamily{famille}`;
- `\fontseries{graisse-largeur}`;
- `\fontshape{forme}`;
- `\fontsize{corps}{interligne}`.

On va détailler ces attributs les uns après les autres et voir comment ils permettent une gestion logique et précise des fontes au moyen des primitives d'affectation ci-dessus ; ensuite on verra comment les commandes de haut niveau concernant le choix des fontes (sections 4.1 et 4.2, pages 23 et 24) sont reliées à ces primitives.

B.3.1 Attribut codage

On donne les codages NFSS les plus couramment utilisés avec une description succincte et un exemple ; bien entendu il y en a beaucoup d'autres :

Table B.1 Attributs de codage NFSS courants

codage	description	exemple
OT1	pour les premières fontes T _E X (CM)	<code>cmr10</code>
T1	codage de Cork, pour les langues européennes mais aussi pour certaines fontes cyrilliques	<code>ec-lmr10</code> <code>wncyr10</code>
T2A,*B,*C	pour l'ensemble des fontes cyrilliques	<code>cmbr10</code>
OML	pour lettres en mode mathématique	<code>cmmi10</code>
OMS	pour les symboles mathématiques	<code>cmsy10</code>
OMX	pour les symboles mathématiques extensibles	<code>cmex10</code>
T4	pour les langues africaines d'alphabet latin	<code>fcr10</code>

L'attribut codage est bien ce qui avait été appelé codage en sortie au début du chapitre 3, page 18 (appel du fichier `fontenc`).

B.3.2 Attribut famille

Les fontes d'une même famille sont des fontes qui ont en commun des caractéristiques graphiques communes, par exemple : pleins et déliés ou non, empattements ou non, chasse constante ou non, ajouré ou non, etc. Les trois familles de fontes pour le texte définies par L^AT_EX sont :

- `cmr` : famille dite romain : A, B, a, b,,
- `cmss` : famille dite sans sérif : A, B, a, b,
- `cmtt` : famille dite machine à écrire (*typewriter*), A, B, a, b.

Bien entendu, on peut en utiliser d'autres, soit en remplacement de celles déjà définies, soit en supplément. La notion de famille, déjà introduite à la section à la section 4.1, page 23, est ainsi définie avec précision.

B.3.3 Attribut graisse-largeur

C'est en quelque sorte un double attribut comprenant deux sous-attributs différents mais partiellement en relation (on comprend très aisément que l'on ne peut pas avoir une fonte très étroitisée avec une forte graisse, c'est-à-dire très grasse) ; on va se borner aux valeurs les plus couramment utilisées :

Table B.2 Attributs de graisse-largeur courants

graisse	valeur	largeur	valeur
light	l	condensed	c
medium (normal)	m	medium (normal)	m
bold	b	expanded	x

Les professionnels de l'impression « grand luxe » disposent de neuf niveaux pour chaque sous-attribut ; L^AT_EX est conçu pour pouvoir les utiliser moyennant l'écriture de quelques commandes supplémentaires. On peut, du point de vue informatique, combiner n'importe quel niveau du premier sous-attribut avec n'importe quel niveau du deuxième groupe ; du point de vue graphique, c'est une autre affaire !

B.3.4 Attribut forme

On donne les formes les plus couramment utilisées (les deux dernières sont moins courantes que les autres) et on les a fait suivre de quelques exemples sur lesquels on remarque que les noms des fontes comprennent les trois caractéristiques graisse-largeur, forme et corps (l'interligne est laissé aux soins du typographe) ; on constatera aussi que les attributs graisse-largeur et forme ne sont pas mentionnés lorsqu'ils ont la valeur dite normale (m, m ou n respectivement).

- `n` droit ou normal ; non mentionné dans les noms des fontes : `cmr10`, `ec-lmr12`, etc.,
- `it` italique ; mentionné par `i` (parfois `ti`) dans les noms des fontes : `cmti10`, `cmbxti12`, `ec-lmri8`, etc.,

- `sl` penché ou oblique ; mentionné par `sl` (parfois `o`, pour oblique) : `cmsl8`, `cmbxsl10`, `ec-lmbxo10`, `ec-lmcsc10`, etc.,
- `sc` petites capitales ; mentionnées par `sc` (les minuscules ne sont pas des capitales réduites géométriquement car il faut garder certaines caractéristiques graphiques, épaisseur des pleins et des déliés par exemple) : `cmbcsc10`, `ec-lmcsc10`, etc.
- `ui` italique droit,
- `ol` éclairée (*outline*).

B.3.5 Attribut corps-interligne

C'est encore un attribut double comprenant deux sous-attributs : la taille proprement dite (le corps, c'est-à-dire la hauteur des capitales de la fonte) et l'interligne (distance entre deux lignes de base consécutives appelée `\baselineskip`). Ces deux sous-attributs ne sont que partiellement dépendants l'un de l'autre (10 pt et 12 pt, 14 pt et 16.5 pt sont des exemples approximatifs) ; en réalité, pour un corps donné, l'interligne qui assure la meilleure lisibilité dépend de la famille considérée⁽⁴⁾.

Un incontestable avantage de l'ensemble de ces conventions et le fait que l'on retrouve en abrégé et avec facilité les attributs des fontes dans les noms de leurs fichiers correspondants ;

B.4 Fontes texte et manipulations élémentaires

La gestion des fontes mathématiques suit les mêmes principes que la gestion des fontes texte ; elle est seulement plus complexe : les caractères sont plus nombreux et, pour une taille de fonte texte donnée, il y a simultanément trois tailles de fontes mathématiques : taille normale, taille d'indice et taille d'indice d'indice (cf. les commandes `\textstyle`), `\displaystyle`, `\scriptstyle` et `\scriptscriptstyle`). On renvoie pour cela le lecteur à la sous-section 7.10.7 de la référence [1].

Pour utiliser une fonte donnée, il faudrait écrire au niveau le plus bas, par exemple :

```
\fontencoding{T1}\fontfamily{lmr}\fontseries{bx}%
\fontshape{it}\fontsize{12pt}{14.5pt}\selectfont
```

ce qui conduit à sélectionner la fonte `ec-lmbxi12` et donne : *essai de fonte*

Heureusement il y a plus simple : ce sont les commandes de haut niveau données. à la section 4.1, page 23. On va donc détailler les relations entre ces commandes de haut niveau et les commandes de bas niveau (commandes d'attributs). A la fin de cette section on aura acquis le nécessaire pour utiliser d'autres fontes.

⁽⁴⁾ Les professeurs de typographie aiment faire l'expérience suivante : ils montrent aux étudiants trois ou quatre paragraphes composés avec une fonte donnée, un corps donné et des interlignes différents ; il montrent ensuite les mêmes paragraphes composés avec une fonte différente : on constate alors que le choix des étudiants pour le meilleur interligne diffère unanimement d'une fonte à l'autre.

B.4.1 Fontes disponibles par défaut

On va présenter sur l'exemple des fontes texte par défaut les principales définitions et commandes de gestion disponibles. Au démarrage de \LaTeX , les paramètres permettant de sélectionner les fontes (d'une manière qui sera expliquée par la suite) prennent les valeurs du tableau B.3 :

Table B.3 Valeurs des attributs des fontes au démarrage de \LaTeX .

Attribut	Valeur	Description
<code>\encodingdefault</code>	<code>OT1</code>	codage fonte principale
<code>\familydefault</code>	<code>\rmdefault</code>	famille fonte principale
<code>\seriesdefault</code>	<code>m</code>	graise fonte principale
<code>\shapedefault</code>	<code>n</code>	forme fonte principale
<code>\rmdefault</code>	<code>cmr</code>	<code>\rmfamily</code> et <code>\textrm</code>
<code>\sfdefault</code>	<code>cmss</code>	<code>\sffamily</code> et <code>\textsf</code>
<code>\ttdefault</code>	<code>cmtt</code>	<code>\ttfamily</code> et <code>\texttt</code>
<code>\bfdefault</code>	<code>bx</code>	<code>\bfseries</code> et <code>\textbf</code>
<code>\mddefault</code>	<code>m</code>	<code>\mdseries</code> et <code>\textmd</code>
<code>\itdefault</code>	<code>it</code>	<code>\itshape</code> et <code>\textit</code>
<code>\sldefault</code>	<code>sl</code>	<code>\slshape</code> et <code>\textsl</code>
<code>\scdefault</code>	<code>sc</code>	<code>\scshape</code> et <code>\textsc</code>
<code>\updefault</code>	<code>n</code>	<code>\upshape</code> et <code>\textup</code>

La colonne « Attribut » est une colonne de paramètres prenant au démarrage de \LaTeX les valeurs données dans la deuxième colonne ; dans la colonne « Description », pour les quatre premières lignes, le texte décrit ce que représente le paramètre correspondant. Pour les lignes suivantes, on a porté les commandes dont l'effet est déterminé par la valeur du paramètre correspondant. Pour exemple, on donne les définitions de trois commandes parmi les neuf :

```
\def\rmfamily{\fontfamily{\rmdefault}\selectfont}
\def\bfseries{\fontseries{\bfdefault}\selectfont}
\def\itshape{\fontshape{\itdefault}\selectfont}
```

Les vraies définitions incluent un peu de code supplémentaire pour différentes raisons ; cependant, cette forme simplifiée produit l'action souhaitée.

Le lecteur est prié de lire encore ce qui suit pour avoir l'explication du rôle des paramètres précédents. Pour simplifier la suite de l'exposé, on ne développe complètement que le cas de la fonte texte principale.

Puisque le codage `OT1` est le codage par défaut (cf. début du fichier maître, chapitre 3) et, puisque l'attribut `\rmdefault` est affecté automatiquement de la valeur `cmr` (nom de la famille principale des fontes CM), \LaTeX va charger le fichier `OT1cmr.fd` qui contient les déclarations de toutes les fontes de la famille `cmr`, c'est-à-dire :

```
\DeclareFontFamily{OT1}{cmr}{}

```

suivie par les déclarations de ses différentes fontes disponibles : on n'en cite que

cinq qui constituent l'ensemble minimal nécessaire à une bonne typographie (droit maigre, droit gras, italique maigre, italique gras et petites capitales droites maigres, il y en a cinq autres peu utilisées) :

```
\DeclareFontShape{OT1}{cmr}{m}{n}{<-> cmr10}{}
\DeclareFontShape{OT1}{cmr}{m}{it}{<-> cmti10}{}
\DeclareFontShape{OT1}{cmr}{bx}{n}{<-> cmbx10}{}
\DeclareFontShape{OT1}{cmr}{bx}{it}{<-> cmbxti}{}
\DeclareFontShape{OT1}{cmr}{c}{sc}{<-> cmcsc10}{}

```

Ensuite, \LaTeX va charger le fichier `OT1cmss.fd` contenant toutes les fontes de la deuxième famille associée :

```
\DeclareFontFamily{OT1}{cmss}{}
dont la première déclaration de fonte est :
\DeclareFontShape{OT1}{cmss}{m}{n}{<-> cmssr10}{}

```

qui correspond au droit maigre mais il y a encore maigre penché, semi-gras droit, gras droit et petites capitales.

Enfin, \LaTeX va charger le fichier `OT1cmtt.fd` contenant toutes les fontes de la troisième famille associée :

```
\DeclareFontFamily{OT1}{cmtt}{}
dont la première déclaration de fonte est :
\DeclareFontShape{OT1}{cmtt}{m}{n}{<-> cmtt10}{}

```

qui correspond au droit maigre mais il y a encore le penché maigre et un italique maigre.

Tout cela concerne le texte, mais il y a aussi tous les fichiers des fontes mathématiques. Pour cela \LaTeX va charger les fichiers `OMLcmr.fd` (fontes des alphabets mathématiques), `OMScmr.fd` (fontes des symboles mathématiques) et `OMXcmr.fd` (fontes des symboles extensibles).

A ce stade, on se rend compte que l'on accède à tout et que l'on peut tout modifier, plus précisément, que les sorciers \LaTeX peuvent tout modifier !

La taille par défaut au démarrage est déterminée par la commande `\normalsize` dont la définition est :

```
\normalsize{\fontsize{10pt}{12pt}}{\selectfont}

```

Ensuite, lorsque l'on écrira le premier mot, la famille principale va être la famille `cmr`, la graisse-largeur va être `m` et la la forme va être `n` : donc la fonte utilisée va être la fonte `cmr10`. Tout cela exige évidemment que tous les fichiers de la fonte en question soient disponibles (fichiers métriques pour la composition et fichier de dessin des caractères pour la visualisation et l'impression) ; mais tout cela est un autre problème qui sera examiné à la section B.5, page 10.

Pour mettre le mot suivant en italique, on écrit d'abord `\itshape`, la forme devient `it` (cf. la définition ci-dessus de cette commande) et c'est la fonte `cmti10` qui est utilisée, les autres attributs restant inchangés.

Pour mettre le mot suivant en gras, on écrit d'abord `\bfseries`, la graisse devient `bx` (cf. la définition ci-dessus de cette commande) et c'est la fonte `cmbx10` qui est

utilisée, à condition d'avoir inhibé l'effet de la commande `\itshape` (mise entre accolades) sinon ce sera la fonte grasse italique `cmbxti10`.

Il reste encore à exposer comment se fait le choix de la taille. Il est en général fait par la classe de document choisie. Pour la classe `book` chargée avec l'option `12pt` (présent document), le fichier de cette classe met en action des définitions des commandes de taille dont on donne seulement trois exemples :

```
\def\small{\fontsize{10pt}{12.3pt}\selectfont}
\def\normalsize{\fontsize{12pt}{14.5pt}\selectfont}
\def\large{\fontsize{14pt}{18.5pt}\selectfont}
```

Elles sont suivies par :

```
\normalsize
```

qui assure que la fonte courante initiale a une taille de 12 pt.

La relation entre les commandes de haut niveau et les commandes de bas niveau est faite. Avant d'aller plus loin, il faut revenir en arrière pour préciser certains points qui ont été simplifiés et parfois omis dans un but pédagogique.

Remarque concernant l'argument taille

Le premier point à préciser est la syntaxe de l'argument de taille (le cinquième) de la déclaration `\DeclareFontShape`. Dans les exemples présentés, cet argument n'est pas très bon car il implique la propriété suivante : quelle que soit la taille choisie (notation $\langle \rightarrow$), les dessins des caractères sont des mises à l'échelle des dessins de la fonte de 10 pt, ce qui est contraire aux règles de la bonne typographie. On montre ci-dessous une vraie fonte de 5 pt⁽⁵⁾ et une vraie fonte de 10 pt (agrandies à 36 pt)

a a

On voit bien que la fonte de 5 pt n'est pas la mise à l'échelle (facteur d'échelle : 0,5) de la fonte de 10 pt !

Dans le cas où l'on dispose d'un jeu de fontes raisonnable, par exemple les fontes de 5, 6, 7, 8, 9, 10, 12 et 17 pt, le cinquième argument peut s'écrire :

```
<5> <6> <7> <8> <9> <10> gen * cmr
<10-11> ec-lmr10 <11-14> cmr12 <14-> cmr17
```

La première ligne signifie que, si l'on choisit une fonte de 5 pt (commande `\tiny`), c'est la fonte `cmr5` qui est utilisée (on a la même propriété pour les cinq autres tailles). La deuxième ligne signifie que, pour chaque intervalle de taille donné entre crochets, `<10-11>` par exemple, c'est la fonte dont le nom suit qui est utilisée après une mise à l'échelle appropriée (le dernier intervalle va jusqu'à l'infini).

En fait, il y a de nombreuses autres possibilités pour écrire ce cinquième argument ; on renvoie le lecteur à la sous-section 7.10.3 de la référence [1] ; on signale seulement qu'il y a aussi la possibilité de déclarer des substitutions : on peut ainsi prévoir la substitution d'une fonte par une fonte aux caractéristiques voisines qui sera utilisée si la fonte choisie n'est pas disponible pour l'utilisateur.

⁽⁵⁾ L'adjectif vrai est pris dans le sens que la fonte a été dessinée pour être utilisée à une certaine taille bien déterminée.

Dernier argument des déclarations

On a remarqué que l'on a laissé vide le troisième (et dernier) argument de la déclaration `\DeclareFontFamily`; par exemple, on peut ainsi changer le caractère de césure des mots (c'est nécessaire pour certaines langues); on peut aussi interdire la césure des mots pour la famille déclarée en prenant pour troisième paramètre :

```
\hyphenchar\font=-1
```

ce qui peut être intéressant pour la famille `cmtt` lorsqu'on l'utilise pour écrire du code informatique.

Le sixième (et dernier) paramètre de la déclaration `\DeclareFontShape` a également été laissé vide; voilà deux exemples de paramètres :

```
\fontdimen6\font=0.9\fontdimen6\font
```

```
\fontdimen1\font=0pt
```

qui modifient les paramètres internes de la fonte : le premier exemple produit une diminution de l'intervalle moyen entre les mots et le deuxième construit une fonte italique droite en redressant une fonte italique normale par l'annulation du déport du haut des capitales. Ce type d'opération, très délicat, est du domaine des \TeX pers.

B.4.2 Ajouter des fontes de la famille utilisée

Dans les distributions, des familles comprennent parfois des fichiers de fontes qu'on ne peut utiliser directement. La famille `lmr` contient les fichiers des fontes `ec-lmb10` (fonte grasse non expensée, `b` au lieu de `bx`⁽⁶⁾) et `ec-lmu10` (italique droit, `u` au lieu de `i`), cf. la section B.3.4 pour la dénomination de la forme. L'utilisation de la première demande la déclaration :

```
\DeclareFontShape{T1}{lmr}{b}{n}{<-> ec-lmb10}{}
```

(déjà faite dans le fichier `T1lmr.fd`) et deux commandes, celle qui définit `\bseries` :

```
\DeclareRobustCommand{\bseries}%
  {\not@math@alphabet\bseries\relax
  \fontseries{b}\selectfont}
```

(la première ligne de la commande est là pour prévenir d'une utilisation de `\bseries` en mode `math`) et enfin, celle qui définit `\textb` à partir de `\bseries` :

```
\DeclareTextFontCommand{\textb}{\bseries}
```

Ces deux commandes doivent être utilisées comme `\bfseries` et `\textbf` :

Essai : gras usuel et nouveau gras avec la commande `\textbf`,

Essai : gras usuel et nouveau gras avec la commande `\textb`.

Pour la deuxième fonte, le travail à faire est le même : la déclaration (encore incluse dans le fichier `T1lmu10.fd`)

```
\DeclareFontShape{T1}{lmu}{m}{ui}{<-> ec-lmu10}{}
```

où `ui` est le code de l'italique droit (cf. la section B.1.4).

Ensuite vient la définition de la commande `\uishape` :

```
\DeclareRobustCommand{\uishape}%
{\not@math@alphabet\uishape\relax\fontshape{ui}\selectfont}
```

suivie de la définition de la commande `\textui` partir de `\uishape` :

⁽⁶⁾ C'est donc une fonte grasse étroitisée idéale pour les titres de section longs.

```
\DeclareTextFontCommand{\textui}{\uishape}
```

Ces deux nouvelles commandes doivent être utilisées comme `\itshape` et `\textit`. Cela donne (on remarquera que l'italique droit n'est pas un simple « redressement géométrique » de l'italique normal) :

Essai d'italique droit avec la commande `\textit`,
 Essai d'italique droit avec la commande `\textui`.

B.4.3 Ajouter une deuxième fonte principale

Si on se trouve dans l'obligation d'écrire de nombreux mot en russe dans un document, on va écrire d'après tout ce qui vient d'être expliqué :

```
\DeclareFontFamily{T1}{cyr}{}
\DeclareFontShape{T1}{cyr}{m}{n}{<-> wncyr10}{}
\DeclareFontShape{T1}{cyr}{m}{it}{<-> wncyi10}{}
\def\russe{\fontfamily{cyr}\selectfont}
```

Si l'on saisit : `{\russe\char"17} \char"6C\char"18`
`\char"62\char"6C\char"18} \char"71\char"61\char"1A}`
 On obtient : Я люблю чай : J'aime le thé.

Bien entendu, pour saisir facilement du russe avec un clavier français, il faudrait avoir un tableau de correspondance. Il faudrait aussi utiliser des fontes de codage T2A comme on l'a mentionné à la section B.1, page 3 : ici on veut simplement montrer que « ça marche » (il se trouve qu'il existe dans la distribution un fichier `t1cyr.fd`, sous forme compactée, qui contient les déclarations ci-dessus).

B.5 Recherche et installation d'une fonte

Dans cette section, on va envisager tous les cas possibles de changement de fontes : depuis le cas où il n'y a que quelques lignes à modifier au cas il faut tout faire à partir des fichiers que l'on a achetés chez l'éditeur. A la fin on aura accédé au grade de petit sorcier L^AT_EX.

Le lecteur comprendra que l'on ne peut pas en quatre ou cinq pages reprendre la bonne centaine de pages de la référence [1] concernant les fontes disponibles (gratuites ou payantes) et le contenu de la référence [3] qui « fait le tour » des fontes mathématiques existantes.

B.5.1 Symboles supplémentaires

On commence par les fontes de caractères spéciaux et d'ornements (en mode texte bien entendu). Il en existe beaucoup, voir [1], pages 387 (symboles PostScrit Zapf Dingbats) et pages 408 à 411. Il y a en particulier les styles (ou packages) `textcomp.sty` et `fourier-orns.sty` qui permettent d'obtenir les symboles monétaires et commerciaux \$, €, ©, etc., les accents très spéciaux : Ü, Â, etc., les symboles d'appel de notes et de paragraphe : ¶, ¶, §, †, ‡, √, etc. ainsi que des symboles d'ornements : ◉, △, ✖, ☞, ●, ☹, ☺, etc. qui rendent un peu vivants les documents. Pour `textcomp.sty`, on trouve la liste complète des symboles disponibles dans [1],

pages 371; pour l'autre style on peut consulter la documentation correspondante `fourier-orns.pdf` de la distribution. Voici des exemples (avec leur commande) :

Table B.4 Quelques symboles spéciaux et leur commande

<code>\textdollar</code>	\$	<code>\textdaggerdbl</code>	‡
<code>\texteuro</code>	€	<code>\textsurd</code>	√
<code>\copyright</code>	©	<code>\noway</code>	⊘
<code>\capitalogonek U</code>	Ū	<code>\danger</code>	⚠
<code>\capitalnewtie A</code>	Â	<code>\decoone</code>	✱
<code>\textpilcrow</code>	¶	<code>\lefthand</code>	☞
<code>\textparagraph</code>	¶	<code>\bomb</code>	💣
<code>\textsection</code>	§	<code>\grimace</code>	😬
<code>\textdagger</code>	†	<code>\aldine</code>	☞

B.5.2 Préparation des tests

On propose au lecteur de tester un certain nombre de fontes : au fur et à mesure des tests, qu'ils soient positifs ou négatifs, on pourra approfondir cette « fabuleuse machinerie » qui assure la gestion des fontes ... quand on sait s'en servir. Pour ces tests, on propose de constituer un répertoire spécial. Dans ce répertoire, on va placer un fichier, nommé `font.tex` contenant diverses commandes pour vérifier le « rendu » des fontes testées :

```

%%%%%%%% font.tex
\documentclass[12pt]{article}
%%%%%%%% Bloc I
.....
%%%%%%%% Bloc II
.....
%%%%%%%%
\begin{document}
{\LARGE\bfseries Titre de chapitre}\par
{\Large\bfseries Section} - \textsf{Test sans sérif}\par
{\large\bfseries Sous-section} - \texttt{Test typewriter}\par
{\bfseries Texte normal gras} - \ae--âêè--\`a\`e\`e\par
{Mots de texte en normal} - \ae--âêè--\`a\`e\`e\par
A B C a b c \textit{A B C a b c} \textbf{A B C a b c}
\textbf{\textit{A B C a b c}} \textsc{A B C a b c}\par
Quelques \textit{mots} pour \textbf{découvrir} la fonte.
$$A=\iint_{\mathbf{B}}^{\boldsymbol{\infty}}f'\left(\max
\left\{\frac{|x^y|}{|x^2+y^2|}\right\}\right)\mathrm{d}x\mathrm{d}y$$
$\A,a$ $\mathrm{A,a}$ $\boldsymbol{A,a}$
$\mathbf{A,a}$ $\mathcal{A}$ $\mathbb{B}$ $\mathsf{A,a}$
$\mathfrak{A}$ - $\ggg$ $\curvearrowleft$ $\circledast$
\end{document}

```

Bien évidemment, le lecteur est encouragé à y ajouter des paragraphes avec différentes tailles, graisses et formes. Voici le résultat pour les fontes LM (style `lmodern`) :

Titre de chapitre

Section - Test sans sérif

Sous-section - Test typewriter

Texte normal gras - æ-àè-àè

Mots de texte en normal - æ-àè-àè

A B C a b c A B C a b c **A B C a b c** **A B C a b c** A B C A B C

Quelques *mots* pour **découvrir** la fonte.

$$A = \iint_{\mathbf{B}}^{\infty} f' \left(\max \frac{\|x^y\|}{|x^2 + y^2|} \right) dx dy$$

AMS : A a A a **A a A a** \mathcal{A} \mathbb{B} A a \mathfrak{A} - \ggg \curvearrowright \otimes

Les tests proposés sont donc (cf. ci-dessus) :

lignes 1, 2, 3 et 4 : gras (de `\LARGE` à `\normalsize`),

lignes 2 et 3 : test de sans sérif et machine à écrire,

lignes 4 et 5 : test ses accents (codages OT1 et T1),

lignes 6 : test de 6 lettres en normal, italique, gras, gras italique et petites capitales,

ligne 7 : une formule,

ligne 8 : test des fontes math, italique, droit, italique gras, droit gras, calligraphique, blackboard, sans sérif, gothique et trois symboles des fontes de l'AMS.

Il est bon de se rappeler que tous les exécutable de la distribution cherchent les fichiers d'abord dans le répertoire courant (le répertoire de travail) puis à l'adresse stockée dans la base de données spécifique. Si l'on veut modifier un fichier, on peut en faire une copie dans le répertoire de travail et modifier seulement la copie (c'est plus facile et surtout sans danger). On peut aussi y placer des fichiers extérieurs à la distribution et que l'on va utiliser provisoirement. Par contre, les fichiers qui doivent compléter la distribution sont à ranger dans l'arborescence après quoi on régénère la base de données (cf. la documentation de la distribution).

B.5.3 Exemples de fontes disponibles

Fontes CM

Bloc : I = vide ; II = pack : `amsmath`, `amssymb`

Le codage par défaut est OT1 et les fontes par défaut sont les CM : il manque donc les lettres accentuées à é è. Par contre il y a celles qui sont saisies à l'aide des macros (ajoutées dès la création de \TeX pour les langues européennes), `\'e`, etc.

Fontes LM

Blocs : I = pack. [T1]fontenc,lmodern ; II = pack. amsmath,amssymb

Le résultat est visible ci-dessus. Le lecteur peut regarder le fichier `lmodern.sty` : au démarrage, la valeur du registre `\rmdefault`, qui a par défaut la valeur `cmr`, prend la valeur `lrm` et idem pour les registres `\sfdefault` et `\ttdefault` qui prennent les valeurs respectives `lmss` et `lmtt`. Les fontes sont ensuite chargées par les fichiers `T1lmr.fd`, `T1lmss.fd` et `T1lmtt.fd`.

Ensuite, pour les mathématiques, les fichiers `OMLlmr.fd`, `OMSlmr.fd` et `OMXlmr.fd` (cf, tableau des codages, page 1), chargent les fontes nécessaires. Enfin, on trouve dans le fichier `lmodern.sty` des définitions relatives aux commandes qui contiennent un argument prenant les valeurs `normal` ou `bold` : cela permet l'utilisation de la commande `\boldmath` qui met les formules en gras. Cette possibilité est très rarement utilisée, sauf peut-être par les enseignants du secondaire (il y a bien d'autres manières plus réussies pour mettre en évidence une formule : fond gris, petit cadre, etc. tout en gardant la possibilité du maigre et du gras pour chaque lettre).

Fontes Fourier-GUTenberg

Blocs : I = pack. [T1]fontenc,fourier ; II = pack. amsmath,amssymb

Ces fontes ont pour famille principale des Utopia dont l'éditeur, Adobe, tolère l'utilisation dans le monde T_EX sans pour cela qu'elles soient dans le domaine public⁽⁷⁾. Par la suite, Michel Bovani, membre de l'association GUTenberg, a complété l'ensemble pour pouvoir disposer de toutes les fontes mathématiques. Les familles associées à la famille principale sont les `cmss` et `cmtt` en codage T1.

La gestion des fontes mathématiques est complètement reprise d'une manière originale donnant la possibilité de diverses options générales décrites dans la documentation correspondante (`fourier-doc-en.pdf`).

Fontes PX

Blocs : I = pack. [T1]fontenc,pxfonts ;
 II = \let\iint\undefined \let\iiint\undefined \let\iiiiint\undefined
 \let\idotsint\undefined + pack. amsmath,amssymb

Ces fontes ont été créées par Young Ruy. On remarque que les formules se lisent aisément (il faudrait un test beaucoup plus important pour bien juger). Avec la distribution de l'auteur du présent document, au début du Bloc II, il a fallu annuler les définitions de quatre commandes déjà définies dans le bloc I (la compilation se fait mais en affichant des messages d'avertissement).

Fontes CM Bright

Blocs : I = pack. [T1]fontenc,cmbright ; II = pack. amsmath,amssymb

⁽⁷⁾ Adobe a aussi publié les documents de référence concernant les langages PostScript et PDF, ce qui a permis le développement de l'ensemble T_EX- \LaTeX .

Créées par Walter Schmidt, ces fontes donnent une impression de légèreté : demanderaient-elles moins d'effort devant une leçon de math de terminale, qui sait ? Elles n'ont pas de petites capitales, ce qui n'est pas un gros défaut pour un manuel scolaire.

Changement de la fonte principale

Pour beaucoup, les mathématiques les mieux réussies sont celles typographiées avec L^AT_EX et l'AMS. Cependant, parmi eux, il y en a qui apprécient le changement de la fonte texte principale : il faut veiller à ce que la fonte remplaçante « s'accorde en un certain sens » avec les fontes utilisées pour les mathématiques. Par exemple, de nombreux ouvrages ont été édités avec des fontes Times (Times Roman et Times New Roman) pour fonte principale : le résultat est excellent. Pour les auteurs et les éditeurs c'est l'assurance de ne pas avoir de surprise ... ce qui est important aujourd'hui où « on n'a pas le temps d'avoir le temps ».

Ce changement est très simple si l'on dispose de tous les fichiers nécessaires (on reviendra sur cette condition plus bas). Par exemple, pour utiliser la famille principale des fontes PX pour fonte principale, il suffit d'écrire :

```
\def\rmdefault{p1xr}
```

où `p1xr` est le nom de cette famille trouvé dans le fichier `pxfonts.sty` : cela est suffisant car L^AT_EX va lire le fichier `t1pxr.fd` et charger les différentes fontes voulues (droit, italique, droit gras, etc.). On peut faire le test avec :

```
Blocs : I = pack. [T1]fontenc + \def\rmdefault{p1xr}
II = pack. amsmath, amssymb
```

Utiliser les fontes de la distribution

Les fontes ont un nom : le choix des noms ne suivent aucune normalisation : les fontes Fourier sont des Utopia adaptées par Michel Bovani ! On a vu les CM, les LM, les PX, les Fourier, les CM Bright et on va en voir d'autres. On va détailler l'arborescence des fichiers de fontes de la distribution sur deux exemples :

Exemple 1 : fontes PX (pxfonts)

Fichiers `.tfm` (fichiers métriques)

```
... \fonts\tfm\public\pxfonts\xxxx.tfm
```

Fichiers `.vf` (fichiers de fontes virtuelles)

```
... \fonts\vf\public\pxfonts\xxxx.vf
```

Fichiers `pfb` (fichiers de dessin des caractères en PostScript)

```
... \fonts\type1\public\pxfonts\xxxx.pfb
```

Fichiers `.map` (transformation des `.pfb` vers les `.vf`)

```
... \fonts\map\dvips\pxfonts\xxxx.map
```

Adobe édite des fichiers de fontes en PostScript qui ont des milliards d'utilisateurs, donc il faut accepter ses choix, codage en particulier. Cela veut dire qu'il faut faire des ré-arrangements du contenu des fichiers `.pfb` pour arriver à des fichiers, nommés fichiers de fontes virtuelles (extension `.vf`), conçues pour être compatibles avec les

normes de L^AT_EX. Ce sont les fichiers d'extension `.map` qui assurent cette tâche⁽⁸⁾. Il y a tous les fichiers nécessaires pour utiliser ces fontes (on les avait testées déjà testées, page précédente).

Exemple 2 : fontes FC (langues africaines d'alphabet latin)

Fichiers `.tfm` (fichiers métriques)

```
... \fonts\tfm\jknappen\cf\xxxx.tfm
```

Il n'y a pas de fichier `.v1` pour les FC, on tente de chercher des fichiers `.mf` dans le sous-répertoire `source` :

```
... \fonts\source\jknappen\fc\xxxx.mf
```

Il y a les fichiers `.mf`, ces fontes sont donc disponibles et on va expliquer comment elles sont exploitées. Le problème a été résolu de la façon suivante : la distribution contient le programme METAFONT qui produit les fichiers `.pk` pour les tailles nécessaires à partir d'un fichier `.mf` (contenant le dessin des caractères) écrit dans un langage semblable au langage PostScript (encore les courbes de Béziérs!). Chaque fois que `pdflatex` (ou `dvips` si on produit le PDF final par l'étape intermédiaire du fichier PostScript, cf. le chapitre 1, page 5) rencontre une déclaration de fonte qui n'a pas de fichier de fonte virtuelle correspondante, il cherche dans le répertoire `source` s'il existe le fichier `.mf` correspondant ; dans le cas positif, il lance METAFONT qui génère les fichiers `.pk` pour toutes les taille nécessaires ; enfin il transforme les données du format `.pk` au format PDF (ou au format PostScript s'il s'agit de `dvips`). Cependant, ces fichiers `.pk` sont stockés sur l'ordinateur : s'ils sont conservés, ils sont utilisés pour les compilations suivantes.

Fontes du domaine commercial

Dans la distribution, il y des fontes pour lesquelles il y a tous les fichiers sauf les fichiers `.pfb` ; il faut les acheter : on verra cela à la section suivante. On prend l'exemple des fontes Times : les fichiers suivants sont disponibles :

```
... \fonts\tfm\adobe\times\xxxx.tfm
```

```
... \fonts\vf\adobe\times\xxxx.vf
```

Il n'y a pas de fichiers `.pfb` ni de fichiers `.mf` mais il y a un fichier `.map` (cf. fin de la section suivante où sera expliqué le rôle de ses fichiers) :

```
... \fonts\map\adobe\times\xxxx.map
```

La présence des fichiers `.map` confirme qu'il y a tout ce qu'il faut à l'exception des fichiers `.pfb` qu'il faut acheter (on verra aussi cela à la prochaine section).

Une modification interne à une famille

Les fontes Concrete paraissent agréables mais elle ont un défaut : il n'y a aucun fichier pour les fontes grasses, par contre, il y a tous les fichiers pour les fontes maigres en codage T1 et en Type 1. Dans la distribution, les fontes manquantes ont été remplacées par des LM : il faut faire le test pour voir l'horreur ;

⁽⁸⁾ Exemple : les petites capitales sont dans un fichier `.pfb` qui ne contient pas les capitales ; mais quand on écrit en petites capitales, on utilise aussi les capitales ... donc il a fallu créer un fichier de fonte (appelée fonte virtuelle) contenant à la fois les capitales et les petites capitales.

Bloc : I = pack. [T1]fontenc, beton, euler ; II = pack. amsmath, amssymb

Il se trouve que les fontes grasses sont disponibles en codage OT1 et sous forme de fichiers .mf. On se propose de les utiliser à la place des LM ; on prend l'exemple du gras droit en déclarant une famille qui n'a qu'une fonte (cf. la déclaration d'une deuxième fonte principale, section B.4.3, page 10) :

```
DeclareFontFamily{OT1}{ccr}{}
Declare\Font\Shape{OT1}{ccr}{b}{n}{%
<11.9>ccb10 <14.6>ccb10 <17.5>ccb10}{}
```

On a choisi **b**, **bx** étant associé aux fontes grasses de remplacement initiales.

La syntaxe de la deuxième ligne de la déclaration de la fonte n'a pas encore été rencontrée. On rappelle qu'il faut un fichier .pk pour chaque taille, donc la syntaxe déjà rencontrée `<->ccb10` n'est pas valable par définition même : il faut demander explicitement toutes les tailles nécessaires, c'est ce qui est fait dans cette ligne. On définit par exemple trois macros :

```
\def\bnor{\fontencoding{OT1}\fontfamily{ccr}
\fontseries{b}\fontshape{n}\fontntsize{11.9}{14.2}\selectfont}
et ensuite \blarge avec 14.6 et 17.6 et et BLarge avec 17.6 et 28.8 (et davantage si
nécessaire). Ces macros s'utilisent comme \normalsize, \large, Large, etc.
```

Pour faire le test, il faut rajouter à la fin du Bloc II les deux déclarations ci-dessus et les trois définitions qui suivent.

Comme cette fonte est codée en OT1, il n'y a que les lettres accentuées saisies avec les commandes du type `\'e` qui sont sorties (on peut contourner la difficulté en intégrant, en début des définitions `\bnor` et suivantes, une macro qui fera le nécessaire pour que `é` soit interprété comme `\'e` et idem pour les autres minuscules accentuées). Par contre les ligatures sont bien rendues. Il manque la fonte texte grasse italique (ou penché), c'est le gras italique des LM qui sort ; ceci est facile réparer en doublant le travail précédent. Pour les mathématiques `\boldsymbol` sort un gras droit (au lieu d'un gras italique), ce qui qui n'est pas gênant quand on le sait ; par contre, `\mathbf` sort la fonte grasse droite des LM qui n'est pas visuellement compatible avec les autres fontes. On peut alors redéfinir la seconde pour qu'elle donne le même résultat que la première.

Il y a une autre possibilité, c'est de prendre les fontes Concrete pour le texte seulement en supprimant le chargement du package `euler` qui gère les fontes mathématiques ; dans ce cas, elles sont gérées par les fontes LM. Cela fait deux sortes de fontes bien différentes : une pour le texte et une pour les formules, ça « passe » bien parce qu'elles ne sont pas mélangées.

B.6 Installer une nouvelle fonte

Dans cette section, on va expliquer en détails l'installation complète d'une fonte dont on n'a pas trouvé la moindre trace dans la distribution. D'abord il va falloir ouvrir le porte-monnaie (plutôt le porte-feuille) !

Exemple : fonte TimesNRMT⁽⁹⁾. On consulte, dans la distribution, le fichier :
`... \doc\fonts\fontname\monotype-fonts.html`
 et l'on trouve cette fonte avec les noms des fichiers à acheter et des renseignements en partie rapportés dans le tableau B.5.

On a aussi pris les fontes Expert adjointes aux quatre fontes de base indispensables car elles seules contiennent les petites capitales et les ligatures ff, fi, fl, ffi et ffl. Chaque fichier d'extension `.pfb` (fichier contenant les dessins des caractères) est accompagné d'un fichier de même nom et d'extension `.afm` qui est le fichier métrique correspondant (fichier contenant les dimensions des caractères). Il est également accompagné d'un fichier d'extension `.inf` qui contient, outre le nom PostScript de la fonte, de nombreuses caractéristiques réservées aux grands sorciers L^AT_EX. Il faut signaler que les fichiers vendus ont le nom complété par autant de caractères `_` qu'il faut pour atteindre 8 caractères : on abandonnera cet artifice en les renommant. Le nom de famille `mntr` a été choisi pour respecter les dénominations de Karl Berry ; les notations `8a` et `8x` sont là pour rappeler les codages initiaux des fontes (codages propres à Adobe) et, pour ne pas s'y perdre, on introduira plus bas la notation `8t` pour identifier les fichiers des fontes virtuelles, de codage T1, que l'on va créer.

Table B.5 Renseignements tirés du fichier `monotype.map`

Nom NFSS du fichier	Nom PostScript	Nom du fichier vendu
<code>mntr8a.pfb</code>	TimesNRMT	<code>tim.pfb</code>
<code>mnti8a.pfb</code>	TimesNRMT-Italic	<code>timi.pfb</code>
<code>mntb8a.pfb</code>	TimesNRMT-Bold	<code>timb.pfb</code>
<code>mntbi8a.afm</code>	TimesNRMT-Bold-Italic	<code>timbi.pfb</code>
<code>mntr8x.pfb</code>	TimesNRExpertMT	<code>tiy.pfb</code>
<code>mnti8x.pfb</code>	TimesNRExpertMT-Italic	<code>tiyi.pfb</code>
<code>mntb8x.pfb</code>	TimesNRExpertMT-Bold	<code>tiyb.pfb</code>
<code>mntbi8x.pfb</code>	TimesNRExpertMT-Bold-Italic	<code>tiybi.pfb</code>

B.6.1 Fichiers de fontes à créer

Pour disposer de l'ensemble minimal de fontes il faut produire les fichiers :

- `mntr8t.tfm` et `mntr8t.vf` : droit maigre,
- `mnti8t.tfm` et `mnti8t.vf` : italique maigre,
- `mntb8t.tfm` et `mntb8t.vf` : gras droit,
- `mntbi8t.tfm` et `mntbi8t.vf` : gras italique,
- `mntrsc8t.tfm` et `mntrsc8t.vf` : petites capitales maigres,

⁽⁹⁾ Cela implique que l'on a consulté les ouvrages, les catalogues, les T_EX-users ayant déjà une bonne expérience, etc.

qui sont les fichiers métriques des différentes fontes (contenant les dimensions, les créneaux, etc.) et les fichiers de fontes virtuelles (que l'on peut caricaturer en disant que leur rôle est de picorer dans les fichiers `.pfb` pour fournir des matériaux réarrangés afin de constituer des fontes virtuelles `.vf` aux normes logiques de \LaTeX). Les fichiers des deuxième, troisième et quatrième couples s'obtiennent comme ceux du premier couple en changeant `r` par `i`, `b` et `bi` : on va se limiter aux premier et cinquième couples.

B.6.2 Programmes utilisés

Le but fixé va être atteint avec trois programmes :

- `fontinst` écrit en langage \TeX que l'on compilera avec l'exécutable `tex` (il est recommandé de lire sa documentation présente dans la distribution),
- `pltotf` transforme les fichiers `.pl` en fichiers `.tfm`,
- `vpltovf` qui génère les fichiers `.tfm` et `.vf` à partir des fichiers `.vpl`.

Le second programme permet de passer d'un fichier lisible par un humain à une forme compactée (on dit aussi compilée) qui peut être rapidement lue par l'exécutable. Il existe des programmes pour faire la transformation inverse. On retrouve ici les principes fondamentaux du monde \TeX : tout est accessible à tous ; par exemple, on peut modifier un créneau entre deux lettres à condition de travailler sur le fichier `.pl`, version lisible du fichier `.tfm`.

B.6.3 Création des fichiers de fontes

On propose au lecteur de faire l'exercice suivant :

- On crée un répertoire `mnr` dans lequel on place les fichiers : `mnr8a.pfb` et `mnr8a.afm`, `mnr8x.pfb` et `mnr8x.afm`, ainsi que les deux fichiers `mnr1.tex` et `mnr2.tex` qui suivent :


```
%%% mnr1.tex
\input fontinst.sty
\installfonts
\transformfont{mnr8r}{\reencodefont{8r}{\fromafm{mnr8a}}}
\endinstallfonts\bye
%%% mnr2.tex
\input fontinst.sty
\installfonts
\installfont{mnr8t}{mnr8r,mnr8x,latin}{T1}{T1}{mnt}{m}{n}{}
\installfont{mnrsc8t}{mnr8r,mnr8x,latinsc}{T1c}{T1}{mnt}{m}{sc}{}
\endinstallfonts\bye
```
- On lance `tex mnr1.tex` pour transformer la fonte `mnr8a` en la fonte `mnr8r` codée en codage `8r` car le codage initial de l'éditeur, `8a`, peut cacher certains caractères (c'est le cas ici). Les fichiers créés sont : `mnr8a.mtx`, `mnr8a.pl`, `mnr8r.mtx` et `mnr8r.pl`.
- On lance `tex mnr2.tex`
 - la première ligne crée la fonte `mnr8t` à partir des fontes `mnr8r` et `mnr8x` ; si des caractères du codage `8t` (ou `T1`) ne sont pas trouvés dans ces deux

fontes ils sont construits avec des caractères de ces deux fontes assemblés par des instructions contenues dans le fichier `latin` : c'est là que l'on réalise la puissance du programme `fontinst` ! Les fichiers créés sont `mnr8x.mtx` et `mntx8x.pl`.

- la deuxième ligne crée la fonte `mnrsc8t` toujours à partir des fontes `mnr8r` et `mnr8x` mais d'une façon différente : les capitales sont prises dans la première et, comme la deuxième contient les petites capitales on dispose ainsi d'une « vraie » fonte de petites capitales avec les capitales. Le lecteur peut ainsi découvrir la raison de l'appellation « fontes virtuelles » : on fait croire à \LaTeX qu'il travaille selon ses possibilités, c'est-à-dire avec une seule fonte à la fois, alors qu'il picore, par l'intermédiaire du fichier d'extension `.vf` dans deux fichiers de fontes `.pfb` à la fois !
- les fichiers créés sont `mnr8x.mtx`, `mnr8x.pl`, `mnr8t.vpl` et `mnrsc8t.vpl` ; en outre, pour le même prix, le programme écrit le fichier `t1mnr.fd` qui est nécessaire pour déclarer les fontes créées avant leur utilisation.
- On écrit enfin, pour avoir les fichiers d'extensions `.vf` et `.tfm` :


```
pltotf mnr8r.pl mnr8r.tfm
pltotf mnr8x.pl mnr8x.tfm
vptovf mnr8t.vpl mnr8t.vf mnr8t.tfm
vptovf mnrsc8t.vpl mnrsc8t.vf mnrsc8t.tfm
```

On dispose d'un fichier tout prêt pour composer un tableau complet des caractères d'une fonte. Arrivé à ce stade on peut lancer :

`latex nfssfont` en répondant par le nom de la fonte, `mnr8t.vf` par exemple, à la première demande et `\table` à la deuxième (si l'on répond `\text` au lieu de `\table`, on aura un paragraphe typographié avec la fonte) puis `\bye` à la dernière.

Le tableau de la fonte est composé mais on ne peut pas encore le voir ni l'imprimer : on a « trompé » \LaTeX mais on a compliqué la vie à `dvips` ; il faut lui préciser quels sont les fichiers dans lesquels il doit aller « picorer » les dessins des caractères en précisant les noms PostScript *exacts* des fontes et leur éventuel recodage.

Pour cela, on écrit le fichier `myfonts.map` suivant :

```
%%% myfonts.map
mnr8r TimesNRMT " TeXBase1Encoding ReEncodeFont " <8r.enc <mnr8a.pfb
mnr8x TimesNRExpertMT <mnr8x.pfb
```

Pour voir le tableau des caractères, on lance alors :

```
dvips -o nfssfont.ps nfssfont.dvi :
```

où le fichier `myfonts.map` indique à `dvips` dans quelles fontes il faut aller chercher les caractères : `mnr8r` (`mnr8a` recodée en code `8r`) et `mnr8x` (non recodée) : c'est bien ce que l'on a fait en lançant `tex fontinst` pour créer les fontes en début de sous-section. Ici, on dispose de la facilité suivante : si `dvips` a trouvé des fontes virtuelles, il cherche un fichier d'extension `.map` portant le nom des fontes, LM, Fourier, PXfonts, etc. En cas d'échec, il cherche un fichier nommé `myfonts.map` ... dans lequel il cherche les noms des fontes intermédiaires produites, `mnr8r` et `mnr8x` dans le cas présent.

Après avoir choisi un nom, TimesNR par exemple, et produit les fichiers `t1mnr.fd`, `mnr8t.tfm`, `mnr8t.vf`, `mnr8r.tfm`, `mnrsc8t.tfm`, `mnrsc8t.vf`, `mnr8x.tfm`, TimesNR.map constitué avec le contenu du fichier `myfonts.map`, il faut les placer, ainsi que les fichiers : `mnr8a.pfb`, `mnr8x.pfb`, dans l'arborescence de la distribution, aux endroits adéquats, par exemple : dans `.../fonts/tfm/adobe/` on crée le répertoire `TimseNR` et l'on y place les fichiers `.tfm` et ainsi de suite.

Il faut ensuite régénérer la base de données comme on le fait lorsque l'on ajoute des paquetages. Il ne faut pas oublier de traiter les deuxième, troisième et quatrième couple de fontes laissés de côté à la sous-section B.6.1, page 17, afin de simplifier l'exposé.

Bien entendu, on peut obtenir directement le tableau des caractères en lançant : `pdflatex nfssfont` (et en répondant aux questions). Ce tableau peut aussi être obtenu pour les fontes disponibles par des fichiers `.pk`, par exemple la fonte `fcr12` (fontes FC, cf. page 15).

Le passage par le fichier PostScript intermédiaire a simplement permis d'expliquer la nécessité et le contenu des fichiers `.map`.

Cette partie II.B de « Débuter en L^AT_EX » se termine sur la méthode pour installer une nouvelle fonte mais il est clair qu'il n'a pas été possible d'aller au fin fond de la machinerie des fontes, en particulier le rôle des fichiers d'extension `mtx` n'a pas été évoqué : c'est dans le fichier `latin.mtx` où l'on peut intervenir pour améliorer certaines ligatures, celles qui sont créées par plusieurs glyphes pris dans une ou plusieurs fontes ; cela est possible car ces fichiers sont lisibles et bien organisés ; le lecteur intéressé est renvoyé à la référence [2] où il peut approfondir beaucoup de choses résumées ci-dessus, en particulier l'utilisation de `fontinst` (pages 273 et suivantes).

Bibliographie

- [1] FRANK MITTELBACH, MICHEL GOOSENS,
L^AT_EX companion, 2^e édition
Pearson Education France, Paris 2005.
- [2] YANNIS HARALAMBOUS,
Fontes & codages,
O'Reilly, Paris 2004.
- [3] THIERRY BOUCHE,
Diversity in math fonts,
TUGboat, 19(2) :120-134,1998.
<http://www.tug.org/TUGboat/Articles/tb19-2/tb59bouc.pdf>

Index

A

attribut, 3

B

`\baselineskip`, 5

C

caractère, adaptation du dessin à la
taille du —, 8

`cmr`, 4

`cmss`, 4

`cmtt`, 4

codage

— 8a, 17

— 8r, 17

— 8t équivalent a T1, 17

— 8x, 17

— des fontes, 3

— OML, 3

— OMS, 3

— OMX, 3

— OT1, 3

— T1, 3

commandes de bas niveau pour utiliser
une fonte, 5

commandes de création de fontes, 18

copyright, 11

corps, 5

corps-interligne, 5

— choix de valeurs du corps et de
l'interligne, 5

courbe de Beziars, 2

D

déclarations complètes pour la famille
`cmr`, 7

`\DeclareFontShape`, 7

`\DeclareRobustCommand`, 9

`\DeclareTextFontCommand`, 9

`\DeclreFontFamily`, 6, 7

`\displaystyle`, 5

dollar, 11

E

euro, 11

F

famille, 4

fichier

— *bitmap*, 2

— *.afm*, 17

— *.fd*, 6

— *.map*, 14

— *.mf*, 15

— *.mtx*, 18, 20

— *.pfa*, 2

— *.pfb*, 2, 14, 17

— *.pk*, 2, 15

— *.pl*, 18

— *.tfm*, 1, 14, 15, 17, 18

— *.vf*, 14, 17, 18

— *.vpl*, 18

— de dessin des caratères, 2

— de test de fontes, 11

— métrique, 1

fichiers de la fonte TimesNR à créer, 17

fichiers de la fonte TimesNR achetés,
17

fonte(s)

— *de type 1*, 2

— *vectorielle*, 2

— CM, 6, 12

— CM Brright, 13

— cyrillique, 10

— Fourier-Gutenberg, 13

— interne, changer une — d'une fa-
mille, 15

— italique droit, 10

- LM, 13
- principale, changement de —, 14
- PX, 14
- recherche des fichiers d'une — commerciale, 15
- recherche des fichiers d'une — de la distribution, 14
- semi condensée, 9

`\fontencoding`, 3

`\fontfamily`, 3

`fontinst`, 18, 20

`fontinst`, 18

`\fontseries`, 3

`\fontshape`, 3

`\fontsize`, 3

forme, 4

- liste de valeurs de l'attribut —, 4

G

graisse-largeur, 4

- tableau de valeurs de l'attribut —, 4

I

impression numérique, 1

in (pouce), 1

interligne, 5

L

`\large`, 8

M

`myfonts.map`, 19

N

`nfssfont.dvi`, 19

`nfssfont.tex`, 19

`\normalsize`, 7, 8

O

OT1, 3

P

`pltopf`, 18

`pltotf`, 18

point d'impression, 1

pt (point), 1

PX, 13

R

rastérisation, 2

relation entre les commandes de haut et bas niveau, 8

résolution, 1

S

`\scriptscriptstyle`, 5

`\scriptstyle`, 5

`\selectfont`, 5

`\small`, 8

symboles spéciaux et ornements, 10

T

T1, 3

taille, 5

`\textstyle`, 5

transformation géométrique d'une fonte, 9

V

valeurs par défaut des attributs des fontes au démarrage de L^AT_EX, 6

`vpltovf`, 18

`vptovf`, 18