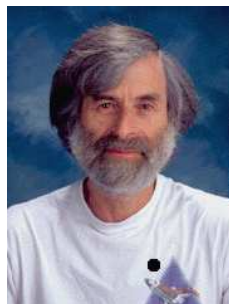


Master MMS et MSID - Première année  
**Introduction à  $\LaTeX$**   
Notes de cours



Don Knuth



Leslie Lamport

FIG. 1 – Les créateurs de  $T_{E}X$  et  $\LaTeX$

Ces notes de cours ont été rédigées à l'aide du livre de F. Mittelbach et M. Goossens,  *$\LaTeX$  Companion* (Pearson Education France, Paris, 2005) ainsi que la Foire Aux Questions sur  $\LaTeX$ <sup>1</sup>. Les définitions de quelques termes techniques proviennent de Wikipédia.

**Table des matières**

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Historique . . . . .	2
1.2	Logiciels . . . . .	3
<b>2</b>	<b>Structure de base d'un fichier <math>\LaTeX</math></b>	<b>4</b>
2.1	Préambule . . . . .	4
2.2	Corps du texte . . . . .	5
2.3	Structure hiérarchique . . . . .	5
2.4	Exemple de base . . . . .	5
2.5	Exercice . . . . .	6
<b>3</b>	<b>Ecrire des mathématiques avec <math>\LaTeX</math></b>	<b>6</b>
3.1	Environnements mathématiques . . . . .	6
3.2	Symboles mathématiques . . . . .	7
3.3	Délimiteurs . . . . .	9
3.4	Ecrire une matrice . . . . .	9
3.5	Gestion des espacements . . . . .	9
3.6	Théorèmes et autres environnements . . . . .	10
3.7	Exercices . . . . .	10

<sup>1</sup>[www.grappa.univ-lille3.fr/FAQ-LaTeX](http://www.grappa.univ-lille3.fr/FAQ-LaTeX)

<b>4</b>	<b>Listes, tableaux et figures</b>	<b>11</b>
4.1	Listes . . . . .	11
4.2	Tableaux . . . . .	11
4.3	Figures . . . . .	13
4.4	Exercices . . . . .	14
<b>5</b>	<b>Inclure une bibliographie</b>	<b>14</b>
5.1	Manière manuelle . . . . .	14
5.2	Manière assistée avec bibtex . . . . .	15
5.3	Quelques règles . . . . .	15
<b>6</b>	<b>Diaporamas et présentations avec L<sup>A</sup>T<sub>E</sub>X</b>	<b>16</b>
6.1	Faire des diaporamas avec beamer . . . . .	16
6.2	Quelques recommandations . . . . .	17
6.3	Exercice . . . . .	17
<b>7</b>	<b>Compléments</b>	<b>17</b>
7.1	Générer une table des matières . . . . .	18
7.2	Générer un index . . . . .	18
7.3	Créer un PDF navigable . . . . .	18
7.4	Gérer les dimensions du document . . . . .	18
7.5	Ecrire une lettre avec L <sup>A</sup> T <sub>E</sub> X . . . . .	19
7.6	Exercices . . . . .	20
<b>8</b>	<b>Ressources électroniques</b>	<b>20</b>
<b>9</b>	<b>Annexe : règles typographiques de la langue française</b>	<b>20</b>

CHAPITRE 1

## Introduction

Après quelques mots sur la genèse de ce système de traitement de texte, on présentera différents logiciels utilisables (et gratuits, ceux payants n'étant pas mentionnés) pour différents systèmes d'exploitation.

### 1.1 Historique

En mai 1977, Donald Knuth de l'Université de Stanford entame des travaux sur un nouveau système de traitement de texte qui deviendra par la suite "T<sub>E</sub>X et Metafont". Initialement, ce système avait été créé uniquement dans un but personnel en vue de la rédaction des trois volumes de *The Art of Computing* (livres fondateurs de l'informatique théorique) qui comprenaient beaucoup d'équations mathématiques. A l'époque, il n'existe que peu de logiciels de traitement de texte satisfaisant pour l'écriture d'équations mathématiques. L'objectif de Don Knuth est élevé : "T<sub>E</sub>X est un nouveau système de composition destiné à la création de beaux livres, particulièrement ceux qui contiennent beaucoup de mathématiques. En préparant un manuscrit au format T<sub>E</sub>X, on ne fait que demander à l'ordinateur de le mettre en page avec une qualité typographique digne des meilleurs imprimeurs". Cet outil devient si important que Gordon Bell n'hésite pas à écrire en 1979 dans la préface d'un livre sur le sujet que " TauEpsilonChi [i.e. T<sub>E</sub>X] de Don Knuth est, dans le domaine de la composition, l'invention la plus marquante de ce siècle. Elle introduit un langage standard en typographie assistée par ordinateur [...]".

Au début des années 1980, Leslie Lamport propose son propre système de composition L<sup>A</sup>T<sub>E</sub>X, basé sur le moteur de composition T<sub>E</sub>X et de son système de macros. En 1985, L<sup>A</sup>T<sub>E</sub>X 2.09 est la première version stable et disponible publiquement. Soutenu par l'American Mathematical Society, L<sup>A</sup>T<sub>E</sub>X est de plus en plus populaire dans la communauté scientifique, académique et industrielle. En 1994, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> devient le nouveau standard.

## 1.2 Logiciels

$\LaTeX$  est un système de traitement de texte qui est très différent des autres logiciels fréquemment utilisés. En effet, les traitements de texte des suites logicielles telles que Open Office ou Microsoft Office sont de type WYSIWYG : What You See Is What You Get. Autrement dit, tout ce qui est saisi à l'écran apparaîtra tel quel sur le papier. Le système  $\LaTeX$  est ensemble d'instructions écrit dans un fichier qui est ensuite compilé à l'aide d'un logiciel spécifique.

**Linux** Sous ce système d'exploitation, un compilateur est déjà installé la plupart du temps par défaut. Pour écrire les instructions, on peut utiliser n'importe quel éditeur de texte. Néanmoins, l'éditeur `emacs` est vivement recommandé. Pour compiler le fichier `MonFichier.tex`, il suffit de lancer la commande `latex MonFichier.tex` dans un terminal. Alors, plusieurs fichiers sont générés dont le fichier `MonFichier.dvi`<sup>2</sup>. On peut le visualiser à l'aide de la commande suivante : `xdvi MonFichier.dvi &`<sup>3</sup>. Le fichier `MonFichier.dvi` permet aussi de générer soit un fichier PostScript<sup>4</sup> soit un fichier PDF<sup>5</sup> : il suffit de lancer respectivement la commande `dvips MonFichier.dvi -o` (si on omet l'option `-o` alors le fichier produit est envoyé à l'imprimante déclarée par défaut) ou `dvipdf MonFichier.dvi`. On peut aussi générer un fichier PDF à partir d'un fichier PostScript à l'aide de la commande `ps2pdf MonFichier.ps`.

On rappelle que pour imprimer un fichier PostScript (ou bien PDF) sous linux, il suffit de lancer la commande `lpr -Pprinter MonFichier.ps` où `printer` est le nom de l'imprimante sur laquelle on veut imprimer le document. Dans le but de faire des économies de papiers, on peut utiliser la commande `psnup` pour transformer un fichier PostScript en nouveau fichier PostScript sur lequel `n` pages seront disposées sur une seule : `psnup -n MonFichier.ps MonFichier2.ps` (en remplaçant `n` par le nombre de pages souhaitées).

**Windows** Sous ce système d'exploitation, on peut utiliser le logiciel gratuit TeXnicCenter<sup>6</sup> (sous licence GPL<sup>7</sup>). Ce logiciel propose outre un compilateur  $\LaTeX$  des raccourcis. Afin de visualiser un fichier PostScript ainsi généré on peut utiliser GhostView<sup>8</sup> et pour un fichier PDF on peut utiliser un fichier Acrobat Reader<sup>9</sup>. Tous ces logiciels sont gratuits et téléchargeables sur internet.

Quel que soit le système d'exploitation utilisé et le logiciel/compilateur utilisé, un fichier  $\LaTeX$  une fois compilé, produit un certain nombre de fichiers. Outre le fichier DVI déjà mentionné, sont produits les fichiers AUX et LOG (journal), ainsi qu'éventuellement les fichiers TOC (tables des matières), IDX (index), LOF (liste des figures), BBL et BLG (bibliographie), etc.

Dans la suite de ce cours, l'accent sera mis sur l'utilisation de  $\LaTeX$  sous linux plutôt que sous Windows. Néanmoins les commandes restent les mêmes que l'on travaille sous Linux ou sous Windows.

---

<sup>2</sup>DVI est un format de fichier ouvert utilisé par le système de composition de texte TeX. DVI est l'abréviation de device-independent, ce qui signifie "indépendant du type d'unité (ou périphérique)". Ainsi, un fichier DVI peut être imprimé sur presque n'importe quel type d'appareil de sortie typographique.

<sup>3</sup>Rappel : l'ajout du symbole `&` permet de lancer une commande en tâche de fond. Ainsi, la commande est toujours exécutée mais on récupère la main pour lancer d'autres commandes.

<sup>4</sup>Le PostScript est un langage essentiellement dédié à la visualisation de pages statiques. Ecrit sous la forme d'un texte ASCII, il est facile à créer, à générer, à comprendre, à transmettre, à manipuler, ...

<sup>5</sup>Le PDF (Portable Document Format) est une version allégée du format PostScript proposée par Adobe. Le PDF s'appuie sur un modèle d'image analogue à PostScript mais avec un format plus structuré afin d'améliorer la performance de visualisation interactive.

<sup>6</sup>[sourceforge.net/projects/texniccenter](http://sourceforge.net/projects/texniccenter)

<sup>7</sup>GPL : licence publique générale GNU, ou GNU General Public License est une licence qui fixe les conditions légales de distribution des logiciels libres du projet GNU.

<sup>8</sup>[pages.cs.wisc.edu/~ghost/gsview](http://pages.cs.wisc.edu/~ghost/gsview)

<sup>9</sup>[www.adobe.com/fr/products/acrobat/readstep2.html](http://www.adobe.com/fr/products/acrobat/readstep2.html)

Un fichier  $\LaTeX$  peut être considéré comme divisé en deux parties distinctes, le préambule et le corps du texte. Le préambule servira essentiellement à donner des informations au compilateur et le corps du texte contiendra le texte lui-même enrichi d'une structure hiérarchique et d'environnements.

A noter dès à présent que des commentaires peuvent être insérés dans le fichier à l'aide du symbole %.

## 2.1 Préambule

Le préambule commence toujours par la commande `\documentclass`. Cette commande contient nécessairement un argument précisant le type de document que l'on rédige : `article`, `book`, `report`, ... Les arguments d'une commande sont toujours spécifiés entre des accolades et les options entre des crochets placées avant l'argument. Parmi les options de la première commande d'un fichier  $\LaTeX$ , on peut préciser la taille de la police, le format de la page, le nombre de colonnes du texte, etc. Par exemple, `\documentclass[10pt,a4paper]{article}` permet de déclarer un document de type `article` avec un police de caractère de 10 pt et sur une feuille au format A4.

Ensuite, s'ensuit tout une liste de déclarations dont l'ordre n'a pas d'importance (mais pour des questions de lisibilité il est préférable de conserver un certain ordre). Parmi ces déclarations, on peut les ranger dans les quatre catégories suivantes :

1. Appels aux "packages"
2. Définitions des marges et espacements
3. Définitions de nouveaux environnements
4. Définitions de macros

L'appel à un package se fait à l'aide de la commande `\usepackage`. Par exemple, afin de spécifier que le document va être écrit en français, on peut faire appel au package `babel` en utilisant l'option `french` : `\usepackage[french]{babel}`. Ce package permet de tenir compte des différentes règles typographiques selon les langues (voir l'annexe sur ce sujet). Les commandes `\selectlanguage{langue}` et `\iflanguage{langue}{expr-vrai}{expr-faux}` permettent d'écrire des documents en plusieurs langues ou pouvant être compilés dans des langues différentes.

Afin de pouvoir écrire le plus facilement possible un texte en français (ou dans d'autres langues), il faut faire appel aux packages `inutenc` et `fontenc` :

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

L'option `latin1` correspond au codage ISO 8859-1 qui permet de représenter la plupart des langues d'Europe occidentale (cela concerne les lettres accentuées en plus de l'alphabet latin) dont le français, l'anglais, l'espagnol, etc. Autrement, il faudrait utiliser les instructions du genre `\'e` pour afficher la lettre é, etc. C'est ce qu'il faut faire dans certains cas lorsque les accents ne figurent pas sur le clavier (par exemple, pour des lettres accentuées dans des langues étrangères).

L'option `T1` du package `fontenc` permet au compilateur d'utiliser le nouveau format d'encodage de fontes. Cette option n'est pas active par défaut pour des raisons de compatibilité avec les versions de  $\LaTeX$  précédentes, mais est à utiliser systématiquement.

Une instruction `\usepackage` peut être utilisée pour une liste de plusieurs packages (sans option) : dans ce cas, il faut préciser les packages séparés par des virgules entre les accolades. Différents packages seront utilisés dans la suite du cours. La plupart de ces packages sont disponibles dans les versions de base de  $\LaTeX$ , mais ils ne sont pas chargés pour ne pas alourdir la compilation.

Si des packages ou des fichiers de style sont manquants (on s'en aperçoit à la compilation par un message d'erreur), il suffit de les trouver sur Internet (en général, il n'y a aucune difficulté à cela) et de les enregistrer dans le répertoire où est le fichier que l'on souhaite compiler.

## 2.2 Corps du texte

Le corps du texte est compris entre les commandes `\begin{document}` et `\end{document}`. On peut commencer le document en précisant un titre, un (ou plusieurs) auteur(s) et une date :

```
\title{La rumeur des steppes. Aral, Asie Centrale, Russie.}
\author{René Cagnat}
\date{\today}
\maketitle
```

La date est alors celle lors de la dernière compilation du document. Si on omet l'argument à l'instruction pour la date, rien n'est affiché et si on veut mettre la date de son choix, il suffit de la préciser dans l'argument.

Certaines parties du corps du texte sont écrites dans ce qu'on appelle un environnement. Tout environnement est signalé par un `\begin` au début et par un `\end` à la fin. Par exemple, cela donne pour l'environnement `equation`, cela donne `\begin{equation}... \end{equation}`.

## 2.3 Structure hiérarchique

Essentiellement, un document est divisé en sections, sous-sections, etc. En fait, le sectionnement dépend du type de document déclaré. Le tableau ci-dessous résume les sectionnements standards d'un fichier  $\LaTeX$ . Les deux derniers niveaux de sectionnement ne sont pas numérotés.

<code>\part</code> (dans book et report)	niveau -1	<code>\part</code> (dans article)	niveau 0
<code>\chapter</code> (dans book et report)	niveau 0	<code>\section</code>	niveau 1
<code>\subsection</code>	niveau 2	<code>\subsubsection</code>	niveau 3
<code>\paragraph</code>	niveau 4	<code>\subparagraph</code>	niveau 5

## 2.4 Exemple de base

Voici pour résumer cette première partie un exemple de base de fichier  $\LaTeX$ . Dans cet exemple, on pourra voir quelques aspects de  $\LaTeX$  non encore mentionnés comme l'effet du symbole `*`, les notes de bas de page, les points de suspension, les caractères spéciaux, ...

```
\documentclass[10pt,a4paper]{article}

\usepackage[french]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}

\begin{document}

\title{La rumeur des steppes. Aral, Asie Centrale, Russie.}
\author{René Cagnat}
\date{\today}
\maketitle

\section*{Introduction. De l'océan des steppes à la mer d'Aral}
```

La fascination de l'Orient qui, d'Alexandre le Grand à Napoléon, a marqué tant d'hommes de caractère et de bons esprits, s'étend aujourd'hui à l'Asie Centrale [\ldots].

```
\section{Première partie. Les beaux restes d'un paradis : initiation à l'Asie Centrale}
```

C'est aussi triste et déconcertant que certains contes orientaux [\ldots].

```

\subsection{Découverte de l'Aral}

''La Mer d'Aral ! Je vous l'assure, je vois la mer d'Aral''. Le chauffeur
ouzbek de la voiture de tête hurle sa joie à la radio [\ldots].

\subsection{Laïka}

{\em Barsakelmès} en kazakh : ''Tu y vas, tu n'en reviens pas''. Tel est
le nom de cette île qui, au c{\oe}ur de la mer d'Aral, est un peu comme
un phare de la désolation [\ldots].

\section{Deuxième partie. Préludes pour un désastre}

[\ldots] Ainsi, le survol de la mer d'Aral\footnote{Les lignes commerciales
Moscou-Tachkent et Francfort-Tachkent permettent de survoler cette zone},
notamment sa côte nord-ouest très escarpée, peut-il donner une idée,
palpable en quelque sorte, de l'incroyable quantité d'eau qui a
disparu [\ldots].
\end{document}

```

## 2.5 Exercice

**Exercice 1** Reproduire le texte ci-dessus. Compiler le fichier et produire les fichiers PostScript et PDF. Visualiser les fichiers DVI, PostScript et PDF.



Afin d'écrire des équations mathématiques, il faut choisir l'environnement mathématique approprié. Différents environnements seront étudiés, puis divers symboles permettant d'écrire des équations.

### 3.1 Environnements mathématiques

Si on souhaite inclure une équation dans un texte, il suffit de l'encadrer à l'aide du symbole  $\$$ . Pour ce qui est d'une équation hors-texte (i.e. une équation mise en évidence en dehors du texte), on a le choix entre plusieurs environnements :

- Le plus simple consiste à encadrer l'équation par les symboles  $\$$  $\$$ . Cela produit toujours une équation non numérotée.
- De manière équivalente, on peut utiliser l'environnement `equation*`. Le symbole `*` permet de supprimer la numérotation. Si omet le symbole `*`, un numéro est affiché automatiquement à droite (par défaut) dans l'ordre chronologique. En général, on numérote uniquement une équation si on fait référence à celle-ci dans le reste du texte. Il faut donc pouvoir faire appel aux équations numérotées dans le texte. Pour cela, il faut utiliser les commandes `\label` et `\ref`. Voici un exemple :

<pre> \begin{equation}\label{eqn:equadif} y' - ay = 0 \end{equation} Les solutions de l'équation différentielle \ref{eqn:equadif} sont de la forme <math>\\$y(x)=ce^{ax}\\$</math> (avec <math>\\$c=y(0)\\$</math>). </pre>	$y' - ay = 0 \tag{1}$
<pre> Les solutions de l'équation différentielle 1 sont de la forme <math>y(x) = ce^{ax}</math> (avec <math>c = y(0)</math>). </pre>	

La commande `\label` permet d'étiqueter une équation et la commande `\ref` permet de faire appel à la numérotation. Ces deux commandes seront utilisées plus loin pour d'autres environnements.

- Pour écrire une succession d'équations, on peut utiliser l'environnement `eqnarray*`. Cet environnement utilise deux points d'alignement symbolisés par un `&`. Chaque ligne se termine par `\\` (sauf la dernière ligne pour laquelle c'est optionnel). Voici un exemple :

<pre>\begin{eqnarray*} Q(x,y) &amp; = &amp; (x-1)(y+1) \\ &amp; = &amp; xy + x - y - 1 \end{eqnarray*}</pre>	$  \begin{aligned}  Q(x,y) &= (x-1)(y+1) \\  &= xy + x - y - 1  \end{aligned}  $
--	--

Comme précédemment, le symbole `*` supprime la numérotation. Si on omet ce symbole, alors tous les lignes sont numérotées. Si on ne souhaite pas numéroter toutes les lignes, on omet le symbole `*` et on rajoute la commande `\nonumber` à chacune des lignes que l'on ne souhaite pas numéroter.

- Une autre manière de procéder consiste à utiliser l'environnement `array` à l'intérieur de l'environnement `equation`. Cela donne par exemple :

<pre>\begin{equation*} \begin{array}{rcl} Q(x,y) &amp; = &amp; (x-1)(y+1) \\ &amp; = &amp; xy + x - y - 1 \end{array} \end{equation*}</pre>	$  \begin{aligned}  Q(x,y) &= (x-1)(y+1) \\  &= xy + x - y - 1  \end{aligned}  $
---	--

Les arguments de l'environnement `array` précisent l'alignement : `r` pour un alignement à droite, `c` pour un alignement centré et `l` pour un alignement à gauche. Cela permet aussi de préciser le nombre de colonnes. Un inconvénient de cet usage est de réduire la taille des équations (même taille qu'une équation dans le texte). Pour y remédier, il faut utiliser la commande `\displaystyle`.

Ces environnements sont disponibles dans la version de base de  $\LaTeX$ . Le package `amsmath` de l'AMS possède des variantes de l'environnement `equation` :

- L'environnement `align` est semblable à l'environnement `eqnarray` mais avec un seul point d'alignement.
- L'environnement `multiline` permet d'écrire une équation sur plusieurs lignes sans alignement.
- L'environnement `split` permet d'écrire une équation sur plusieurs lignes avec alignement.

Pour encadrer une équation, on peut utiliser la commande `\boxed` dans un environnement mathématique.

### 3.2 Symboles mathématiques

Avant d'énumérer différents types de symboles mathématiques, voici deux commandes importantes : celles pour les indices et celles pour les exposants. Le symbole `_` permet d'écrire du texte en indice et le symbole `^` permet d'écrire du texte en exposant. Ces instructions permettent de mettre une lettre ou un symbole en indice ou exposant. Pour un groupe de texte et/ou de symboles, il faut l'encadrer d'accolades. Par exemple, on peut écrire `x_{ij}=1` ou `y=x^2`. Cela donne respectivement les résultats suivants :  $x_{ij} = 1$  et  $y = x^2$ .

Ci-dessous, on trouvera plusieurs listes de symboles utilisés en mathématiques. Ces listes sont présentées par catégories. Certains de ces symboles nécessitent l'utilisation des packages suivants : `latexsym`, `amssymb` et `amsmath`.

## Fonctions et opérateurs prédéfinis

<code>\arcsin</code>	arcsin	<code>\arg</code>	arg	<code>\cos</code>	cos	<code>\coth</code>	coth
<code>\deg</code>	deg	<code>\det</code>	det	<code>\dim</code>	dim	<code>\exp</code>	exp
<code>\inf</code>	inf	<code>\ker</code>	ker	<code>\lim</code>	lim	<code>\liminf</code>	lim inf
<code>\limsup</code>	lim sup	<code>\ln</code>	ln	<code>\max</code>	max	<code>\min</code>	min
<code>\sin</code>	sin	<code>\sinh</code>	sinh	<code>\sup</code>	sup	<code>\tan</code>	tan
<code>\varliminf</code>	$\varliminf$	<code>\varlimsup</code>	$\varlimsup$	<code>\sum</code>	$\sum$	<code>\prod</code>	$\prod$
<code>\int</code>	$\int$	<code>\iint</code>	$\iint$	<code>\iiint</code>	$\iiint$	<code>\idotsint</code>	$\int \cdots \int$
<code>\sqrt{}</code>	$\sqrt{\quad}$	<code>\frac{ }{ }</code>	$\frac{\quad}{\quad}$				

Certains de ces symboles nécessitent le package `amsmath`. C'est le cas par exemple des symboles  $\varliminf$  et  $\varlimsup$ .

## Symboles ordinaires

<code>\imath</code>	$i$	<code>\jmath</code>	$j$	<code>\mathdollar</code>	$\$$	<code>\bot</code>	$\perp$
<code>\backslash</code>	$\backslash$	<code>\emptyset</code>	$\emptyset$	<code>\exists</code>	$\exists$	<code>\forall</code>	$\forall$
<code>\infty</code>	$\infty$	<code>\neg</code>	$\neg$	<code>\mathbb{N}</code>	$\mathbb{N}$	<code>\mathbb{R}</code>	$\mathbb{R}$

**Symboles de relation** Ci-dessous quelques symboles de relation sont présentés.

<code>\cup</code>	$\cup$	<code>\cap</code>	$\cap$	<code>\div</code>	$\div$	<code>\pm</code>	$\pm$
<code>\land</code>	$\wedge$	<code>\lor</code>	$\vee$	<code>\setminus</code>	$\setminus$	<code>\star</code>	$\star$
<code>\odot</code>	$\odot$	<code>\oplus</code>	$\oplus$	<code>\otimes</code>	$\otimes$	<code>\cdot</code>	$\cdot$
<code>&lt;</code>	$<$	<code>\leq</code>	$\leq$	<code>\leqslant</code>	$\leqslant$	<code>=</code>	$=$
<code>&gt;</code>	$>$	<code>\geq</code>	$\geq$	<code>\geqslant</code>	$\geqslant$	<code>\approx</code>	$\approx$
<code>\asymp</code>	$\asymp$	<code>\equiv</code>	$\equiv$	<code>\gg</code>	$\gg$	<code>\ll</code>	$\ll$
<code>\prec</code>	$\prec$	<code>\preceq</code>	$\preceq$	<code>\succ</code>	$\succ$	<code>\succeq</code>	$\succeq$
<code>\sim</code>	$\sim$	<code>\simeq</code>	$\simeq$	<code>\neq</code>	$\neq$	<code>\in</code>	$\in$
<code>\ni</code>	$\ni$	<code>\notin</code>	$\notin$	<code>\subset</code>	$\subset$	<code>\subseteq</code>	$\subseteq$
<code>\supset</code>	$\supset$	<code>\supseteq</code>	$\supseteq$	<code>\mapsto</code>	$\mapsto$	<code>\leadsto</code>	$\leadsto$
<code>\leftarrow</code>	$\leftarrow$	<code>\Leftarrow</code>	$\Leftarrow$	<code>\rightarrow</code>	$\rightarrow$	<code>\Rightarrow</code>	$\Rightarrow$
<code>\Lrightarrow</code>	$\Lrightarrow$	<code>\Rrightarrow</code>	$\Rrightarrow$	<code>\uparrow</code>	$\uparrow$	<code>\downarrow</code>	$\downarrow$
<code>\perp</code>	$\perp$	<code>\models</code>	$\models$	<code>\mid</code>	$\mid$	<code>\propto</code>	$\propto$

Pour écrire au-dessus et/ou au-dessous d'une simple flèche, on peut utiliser par exemple la commande `\xrightarrow[bas]{haut}` où `bas` est le texte qui sera écrit au-dessous de la flèche et `haut` le texte qui sera écrit au-dessus de la flèche.

**Lettres grecques et autres symboles** Toutes les lettres grecques ne sont pas disponibles en majuscule. Ci-dessous les lettres grecques les plus employées sont répertoriées.

<code>\Delta</code>	$\Delta$	<code>\Gamma</code>	$\Gamma$	<code>\Lambda</code>	$\Lambda$	<code>\Omega</code>	$\Omega$
<code>\Phi</code>	$\Phi$	<code>\Psi</code>	$\Psi$	<code>\Sigma</code>	$\Sigma$	<code>\Theta</code>	$\Theta$
<code>\alpha</code>	$\alpha$	<code>\beta</code>	$\beta$	<code>\chi</code>	$\chi$	<code>\delta</code>	$\delta$
<code>\epsilon</code>	$\epsilon$	<code>\eta</code>	$\eta$	<code>\gamma</code>	$\gamma$	<code>\kappa</code>	$\kappa$
<code>\lambda</code>	$\lambda$	<code>\mu</code>	$\mu$	<code>\nu</code>	$\nu$	<code>\omega</code>	$\omega$
<code>\phi</code>	$\phi$	<code>\pi</code>	$\pi$	<code>\psi</code>	$\psi$	<code>\rho</code>	$\rho$
<code>\sigma</code>	$\sigma$	<code>\tau</code>	$\tau$	<code>\theta</code>	$\theta$	<code>\upsilon</code>	$\upsilon$
<code>\xi</code>	$\xi$	<code>\zeta</code>	$\zeta$	<code>\varepsilon</code>	$\varepsilon$	<code>\varphi</code>	$\varphi$
<code>\varpi</code>	$\varpi$	<code>\varrho</code>	$\varrho$	<code>\varsigma</code>	$\varsigma$	<code>\vartheta</code>	$\vartheta$

**Ponctuations** Les symboles de ponctuation sont très utiles en mathématiques pour écrire des systèmes d'équations, des matrices, etc.

<code>\codts</code>	$\cdots$	<code>\ldots</code>	$\dots$
<code>\ddots</code>	$\ddots$	<code>\vdots</code>	$\vdots$



**Accents mathématiques** Il est possible de rajouter des accents pour les variables d'une équation. Le tableau ci-dessous résume quelques accents mathématiques les plus souvent utilisés :

<code>\bar{x}</code>	$\bar{x}$	<code>\hat{x}</code>	$\hat{x}$	<code>\widehat{x}</code>	$\widehat{x}$
<code>\dot{x}</code>	$\dot{x}$	<code>\ddot{x}</code>	$\ddot{x}$	<code>\vec{x}</code>	$\vec{x}$
<code>\tilde{x}</code>	$\tilde{x}$	<code>\widetilde{x}</code>	$\widetilde{x}$	<code>\check{x}</code>	$\check{x}$

### 3.3 Délimiteurs

On appelle délimiteurs tout symbole tel que les parenthèses, les crochets, les accolades, mais aussi les flèches, etc. Afin d'ajuster de manière automatique la hauteur des délimiteurs, on peut utiliser les commandes `\left` et `\right`. Le tableau ci-dessous indique quelques délimiteurs possibles :

<code>( )</code>	<code>()</code>	<code>\{ \}</code> ou <code>\lbrace \rbrace</code>	<code>{ }</code>	<code>[ ]</code> ou <code>\lbrack \rbrack</code>	<code>[]</code>
<code>\langle \rangle</code>	<code>\langle \rangle</code>	<code>\vert</code> ou <code> </code>	<code> </code>	<code>\vert</code> ou <code>\ </code>	<code>\ </code>
<code>\lceil \rceil</code>	<code>\lceil \rceil</code>	<code>\lfloor \rfloor</code>	<code>\lfloor \rfloor</code>	<code>\sqrt</code>	<code>\sqrt</code>

On est obligé d'utiliser les deux délimiteurs en même temps. On ne peut pas utiliser l'un sans l'autre, mais si on ne souhaite pas utiliser un délimiteur à droite on le remplace par un point (.). C'est le cas par exemple si on souhaite écrire un système d'équation (en général, on ne met qu'une accolade à gauche) :

<pre>\begin{equation*} \left\{ \begin{array}{rcl} Q(x,y) &amp; = &amp; (x-1)(y+1) \\ &amp; = &amp; xy + x - y - 1 \end{array} \right. \right.</pre>	$\left\{ \begin{array}{l} Q(x,y) = (x-1)(y+1) \\ = xy + x - y - 1 \end{array} \right.$
---	--

### 3.4 Ecrire une matrice

Pour écrire une matrice (en hors-texte), on peut combiner les environnements `equation` et `array` comme expliqué ci-dessus. Autrement, on peut utiliser l'environnement `matrix`. Cet environnement ne met pas de délimiteurs. Pour mettre des délimiteurs (les mêmes à gauche et à droite), il faut utiliser l'environnement approprié : `pmatrix` (parenthèse), `bmatrix` (crochet), `Bmatrix` (accolade), `vmatrix` (trait vertical) ou `Vmatrix` (double traits verticaux). Dans tous les cas, il n'y a pas besoin de préciser le nombre de colonnes (contrairement à l'environnement `array`). Néanmoins on est limité à dix colonnes.

### 3.5 Gestion des espacements

Il est possible d'ajouter des espaces, positifs ou négatifs, dans des équations mathématiques. Le tableau ci-dessous résume les différentes commandes pour les différents espacements possibles :

Espacements positifs			Espacements négatifs		
Espace	Commande	Raccourci	Espace	Commande	Raccourci
→ ←	<code>\thinspace</code>	<code>\,</code>	← →	<code>\negthinspace</code>	<code>\!</code>
→ ←	<code>\medspace</code>	<code>\:</code>	← →	<code>\negmedspace</code>	
→ ←	<code>\thickspace</code>	<code>\;</code>	← →	<code>\negthickspace</code>	
→ ←	<code>\enspace</code>				
→ ←	<code>\quad</code>				
→ ←	<code>\qquad</code>				

### 3.6 Théorèmes et autres environnements

Pour définir de nouveaux environnements tels que les théorèmes, propositions, définitions, etc., il faut les déclarer dans le préambule à l'aide de la commande `\newtheorem{nom}{texte}` qui contient deux arguments. A chaque environnement ainsi créé, un compteur est associé. A chaque utilisation de l'environnement nouvellement défini,  $\LaTeX$  imprime le second argument (i.e. `texte`) suivi du numéro. Le texte contenu dans l'environnement est alors écrit en italique. Par exemple, si on déclare dans le préambule l'environnement `\newtheorem{thm}{Théorème}`, alors dans le corps du texte on peut écrire le théorème suivant :

```
\begin{thm}
Une partie  $H$  non vide du groupe
 $G$  est un sous-groupe de  $G$  si,
et seulement si,
\begin{equation*}
\forall (x,y) \in H^2 \ ,
\quad x \star y^{-1} \in H \ ;.
\end{equation*}
\end{thm}
```

**Théorème 1** Une partie  $H$  non vide du groupe  $G$  est un sous-groupe de  $G$  si, et seulement si,

$$\forall (x,y) \in H^2 \quad x \star y^{-1} \in H .$$

On peut aussi définir de nouveaux environnements à l'aide de la commande `\newenvironment` : le premier argument contient le nom de ce nouvel environnement, le deuxième (optionnel) indique le nombre d'arguments, le troisième la définition de début d'environnement et le dernier la définition de fin d'environnement. Par exemple, pour définir un environnement pour les preuves (d'un théorème, d'une proposition, etc), on peut utiliser le nouvel environnement suivant :

```
\newenvironment{demo}{
\noindent {\bf Démonstration} \noindent} {\hfill  $\Box$ \vskip 5mm}
```

Son utilisation est simple. Voici un exemple :

```
\begin{demo}
La preuve de ce théorème est
immédiat.
\end{demo}
```

**Démonstration** La preuve de ce théorème est immédiat.  $\square$

### 3.7 Exercices

**Exercice 2** Reprendre tous les exemples ci-dessus et les inclure dans un fichier  $\LaTeX$ . Visualiser les fichiers obtenus.

**Exercice 3** Ecrire un système linéaire de deux équations à deux inconnues. Proposer plusieurs manières de l'écrire en  $\LaTeX$ .

**Exercice 4** Ecrire le système linéaire de l'exercice précédent à l'aide d'une matrice. Proposer plusieurs manières de l'écrire en  $\LaTeX$ .

**Exercice 5** Ecrire la démonstration du résultat suivant :

$$I = \int_0^{\infty} e^{-x^2/2} dx = \frac{1}{2} \sqrt{2\pi} .$$

On rappelle qu'une démonstration possible consiste à calculer  $I^2$  comme une intégrale double en passant aux coordonnées polaires.

**Exercice 6** Ecrire en  $\LaTeX$  le théorème de Pythagore.

Le système  $\LaTeX$  permet de générer facilement des listes et des tableaux, ainsi que d'inclure des figures.

## 4.1 Listes

Il faut distinguer les listes numérotées des listes non numérotées. Pour une liste numérotée, il faut utiliser l'environnement `enumerate` et pour une liste non numérotée l'environnement `itemize`. Dans les deux cas, les éléments de la liste sont générés par la commande `\item`. Ce sont les deux styles de base, il en existe d'autres propres à certains packages. Voici un exemple de liste numérotée :

<pre>\begin{enumerate} \item Nb étudiants en M1 MMS : 12 \item Nb étudiants en M1 MSID : 11 \end{enumerate}</pre>	<ol style="list-style-type: none"> <li>1. Nb étudiants en M1 MMS : 12</li> <li>2. Nb étudiants en M1 MSID : 11</li> </ol>
---	---

Et voici un exemple de liste non numérotée :

<pre>\begin{itemize} \item Nb étudiants en M1 MMS : 12 \item Nb étudiants en M1 MSID : 11 \end{itemize}</pre>	<ul style="list-style-type: none"> <li>– Nb étudiants en M1 MMS : 12</li> <li>– Nb étudiants en M1 MSID : 11</li> </ul>
---	---

## 4.2 Tableaux

Il est possible de créer des tableaux qui seront inclus dans le texte. Pour cela, il faut utiliser l'environnement `tabular`. En argument, on précise l'alignement de chaque colonne (`l`, `c` ou `r` : voir l'environnement `array`) et donc le nombre de colonnes. Le séparateur de colonne est symbolisé par `&` et alfin de ligne par `\\`.

Si on souhaite rajouter des séparations entre deux colonnes, il suffit de placer à l'endroit souhaité le symbole `|` dans les arguments dans l'environnement. Pour rajouter des séparations entre deux lignes, il suffit d'utiliser la commande `\hline` comme dans l'exemple ci-dessous.

Par défaut, le tableau est aligné à gauche. Si on souhaite que celui-ci soit centré, il suffit d'encapsuler le tableau dans l'environnement `center`. De plus, si on souhaite donner un titre au tableau et/ou numéroter le tableau, il faut utiliser l'environnement `table` et y encapsuler le tableau. Ensuite, dans cet environnement, on peut respectivement les commandes `\label` et `\caption`. Pour utiliser le numéro du tableau dans le texte, il suffit d'utiliser la commande `\ref`. L'inconvénient de l'usage de ce tableau est une mauvaise maîtrise du positionnement dans le texte. Pour y remédier partiellement, l'environnement `table` prend en option la position du tableau dans le texte : `h` pour le placer à l'endroit où il est situé, `t` pour le placer en haut de la page et `b` pour le placer en bas de la page. On peut mettre en option un, deux ou trois caractères et le compilateur essaiera de le placer dans l'ordre spécifié.

Voici un exemple relativement complet<sup>10</sup> :

```
\begin{table}[htp]
\begin{center}
\begin{tabular}{|cc|c|}
\hline
$x$ & $y$ & $\text{and}(x,y)$ \\
\hline
```

<sup>10</sup>Il ne m'est pas possible de montrer le résultat comme pour les exemples précédents car l'objet est créé est un *flottant* qu'il n'est pas possible d'insérer un flottant dans l'environnement `multicols` utilisé pour écrire du texte sur deux colonnes. Une autre solution devra être employée...

```

0 & 0 & 0 \\
\hline
1 & 0 & 1 \\
\hline
0 & 1 & 1 \\
\hline
1 & 1 & 1 \\
\hline
\end{tabular}
\end{center}
\label{tab:images_and_1}
\caption{La fonction logique  $\$and\$$ }
\end{table}

```

Le package `multirow` permet de fusionner les colonnes ou les lignes d'un tableau. Il faut utiliser les commandes `\multicolumn` et `\multirow`. Pour la commande `\multicolumn`, le premier argument indique le nombre de colonnes à fusionner, le deuxième l'alignement avec les éventuels délimiteurs et le troisième le texte. Cela donne par exemple :

```

\begin{center}
\begin{tabular}[b]{|l|c|}
\hline
\multicolumn{2}{|c|}{Texte} \\
\hline \hline
donnée 1 & donnée 2 \\
A & B \\
\hline
\end{tabular}
\end{center}

```

Texte	
donnée 1	donnée 2
A	B

A noter que, dans le cas où le nombre de colonnes à couvrir est égal à 1, cette commande peut être utilisée pour modifier l'alignement (`c`, `l`, `r`, `|`, etc) d'une cellule.

La commande `\multirow` est un peu plus complexe à décrire et à utiliser. Le plus simple est d'essayer de comprendre l'exemple ci-dessous :

```

\begin{center}
\begin{tabular}{|c|c||c|c|}
\hline
\multirow{2}{0.5cm}{k}
& \multirow{2}{0.5cm}{ $p_G$ }
& \multicolumn{2}{|c|}{test} \\
\cline{3-4}
& & DADWRD & RARWRD \\
\hline
2 &  $1$  & 90 n & 228 n \\
3 &  $p_d$  & 202 n & 449 n \\
4 &  $p_d^2$  & 424 n & 891 n \\
5 &  $p_d^3$  & 866 n & 1774 n \\
\hline
\end{tabular}
\end{center}

```

k	$p_G$	test	
		DADWRD	RARWRD
2	1	90 n	228 n
3	$p_d$	202 n	449 n
4	$p_d^2$	424 n	891 n
5	$p_d^3$	866 n	1774 n

Enfin, le package `\slashbox` permet de diviser une cellule en deux. Pour cela, il faut utiliser la commande `\backslashslashbox{sud-ouest}{nord-est}`. En reprenant l'exemple ci-dessus, cela donne :

```

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\backslashslashbox{$x$}{$y$} & 0 & 1 \\
\hline
0 & 0 & 0 \\
\hline
1 & 0 & 1 \\
\hline
\end{tabular}
\end{center}

```

	$y$	0	1
$x$	0	0	0
1	0	0	1

Si un tableau est trop grand pour tenir en mode portrait, on peut changer son orientation en l'encapsulant à l'aide de la commande `\rotatebox`. Cette commande prend deux arguments : dans la première accolade, on indique le degré de rotation (90 pour que le tableau soit en mode paysage) et dans la seconde accolade les instructions pour la construction du tableau.

### 4.3 Figures

Pour insérer une image dans un fichier  $\LaTeX$ , il est plus simple de disposer d'un fichier au format EPS (Encapsulated PostScript). Pour convertir une image au format EPS, on peut utiliser un logiciel de traitement d'image tel que le logiciel gratuit `GIMP` (disponible sous linux et windows). Pour créer une figure ou un diagramme, on peut utiliser, sous linux, le logiciel `xfig` (en général, installé dans les distributions de base). Certains logiciels tels que `Scilab` peuvent produire des figures au format EPS voire au format FIG (celui du logiciel `xfig`), ces figures pouvant être ensuite retravaillées.

L'insertion du package `graphicx` permet d'inclure des images à l'aide de la commande `\includegraphics[width=8cm]{MonImage.eps}`. L'option `width` permet de fixer la largeur, alors que l'option `height` permet de fixer la longueur.

De même que pour un tableau, si on souhaite que l'image soit centrée, il faut utiliser la commande `\includegraphics` dans l'environnement `center`. Pour numéroter l'image et/ou donner un titre, il faut utiliser respectivement les commandes `\label` et `\caption` dans l'environnement `figure` (qui prend les mêmes options que l'environnement `table`).

Les packages `psstricks`, `pst-node` et `pst-text` et `pst-3d` permettent de remplacer du texte par des équations en  $\LaTeX$ . Pour cela, il faut utiliser la commande `\psfrag`. Voici un exemple relativement complet :

```

\begin{figure}[htp!]
\begin{center}
\psfrag{a}[][][{$a_j$}]
\psfrag{b}[][][{$\mbox{Me}$}]
\psfrag{c}[][][{$a_{j\check{1}}$}]
\psfrag{d}[][][{$F(a_j)$}]
\psfrag{e}[][][{$0.5$}]
\psfrag{f}[][][{$F(a_{j\check{1}})$}]
\includegraphics[width=5cm]{mediane.eps}
\end{center}
\caption{Localisation de la médiane}
\label{fig:mediane}
\end{figure}

```

Le package `subfig` permet facilement de créer un tableau de figures, chacune des figures ayant son label et son sous-titre. Voici un exemple d'utilisation :

```

\begin{figure}\centering
\subfloat[Year 1987]
{\includegraphics[width=7cm]{menard87.eps}\label{fig:m1987}}

```

```

\qqquad
\subfloat[Year 1988]
{\includegraphics[width=7cm]{menard88.eps}\label{fig:m1988}}
\\
\subfloat[Year 1989]
{\includegraphics[width=7cm]{menard89.eps}\label{fig:m1989}}
\qqquad
\subfloat[Year 1990]
{\includegraphics[width=7cm]{menard90.eps}\label{fig:m1990}}
\caption{Weekly planktonic abundance for four years}
\label{fig:menard}
\end{figure}

```

## 4.4 Exercices

**Exercice 7** Ecrire un formulaire de trigonométrie.

**Exercice 8** Ecrire un formulaire d'identités remarquables.

**Exercice 9** Ecrire le triangle de Pascal jusqu'au rang 5.

**Exercice 10** Choisir une fonction  $f$  (par exemple, un polynôme de degré 3) et écrire en  $\text{\LaTeX}$  son tableau de variation.

**Exercice 11** Représenter graphiquement la convergence de la suite  $u_n = u_{n-1}^2$  en partant du point initial  $x_0 = \frac{1}{4}$ . Inclure cette figure dans un document  $\text{\LaTeX}$ .

**Exercice 12** Représenter graphiquement l'interpolation linéaire de la fonction  $x \mapsto \frac{1}{x}$  entre les points  $x_1$  et  $x_2$ . Inclure cette figure dans un document  $\text{\LaTeX}$ . On pourra utiliser le package `pstricks`.

**Exercice 13** Énoncer le théorème de Thalès et l'illustrer par un graphique qui sera inséré après l'énoncé du théorème.

**Exercice 14** Choisissez un de vos cours et rédigez, à l'aide  $\text{\LaTeX}$ , une partie de celui-ci.

CHAPITRE 5

## Inclure une bibliographie

Dans le but de produire une bibliographie, deux possibilités se présentent : on pourra l'effectuer soit de manière manuelle, soit de manière assistée à l'aide de `bibtex`. Pour mettre à jour la bibliographie, il faut compiler deux fois le fichier source, quelque soit la manière employée.

### 5.1 Manière manuelle

Il existe un environnement standard pour produire une liste de références bibliographiques, `\thebibliography`. Chaque élément de la bibliographie commence par la commande `\bibitem`. Pour séparer les différents éléments d'une notice, on peut utiliser la commande `\newblock`. Voici un exemple :

```

\bibliographystyle{plain}
\begin{thebibliography}{2}
\addcontentsline{toc}{section}{Références}

\bibitem{calot73}
G. Calot.
\newblock {\em Cours de statistique descriptive}.
\newblock Dunod décision, 1973.

```

```

\bibitem{simpson51}
E.H. Simpson.
\newblock The interpretation of interaction in contengency tables.
\newblock {\em J. Roy. Statist. Soc (ser. B)}, 13\string: 238--241, 1951.

\end{thebibliography}

```

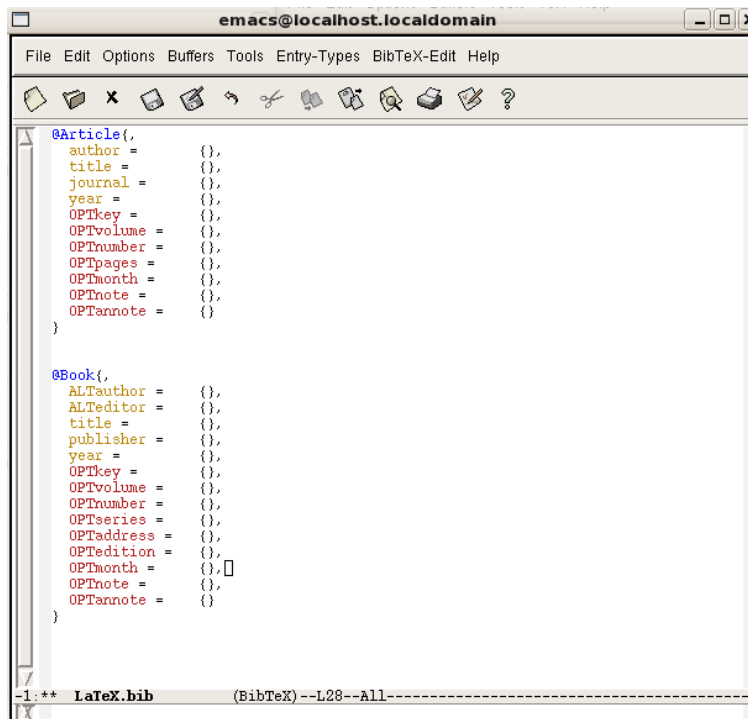
L'argument de l'environnement précise le nombre maximal d'entrées dans la bibliographie. On notera que `{\em ...}` permet d'écrire du texte en italique et que la commande `\string` permet de supprimer l'espace (utilisé ici pour un usage de la typologie anglaise, le journal étant de cette langue).

Avant le début de l'environnement, on peut déclarer un style pour la bibliographie à l'aide de la commande `\bibliographystyle:plain,alpha,chicago` et `natbib` sont des styles bibliographiques possibles.

Pour citer une référence dans le texte, il suffit d'utiliser la commande `\cite`. Par exemple : `Simpson \cite{simpson51}` a étudié ce paradoxe dans son article.

## 5.2 Manière assistée avec bibtex

Pour créer un fichier de bibliographie, on peut aussi utiliser `bibtex` pour créer un fichier au format BIB. Dans ce cas, on recommande l'utilisation de l'éditeur de texte `emacs` sous linux. En effet, `emacs` s'adapte en fonction de l'extension du fichier invoqué lors du lancement de la commande. Pour saisir des entrées, il suffit de choisir le type de références dans le menu `Entry-Types` : les champs ayant le préfixe `OPT` sont optionnels et ceux ayant le préfixe `ALT` sont des choix alternatifs (exclusifs). Une capture d'écran est présentée ci-dessous. Une fois le fichier enregistré, il faut compiler la bibliographie du fichier `LaTeX` avec la commande `bibtex MonFichier`. Cela produit plusieurs fichiers dont le fichier `BBL` qui contient une liste bibliographique au même format que ce qui peut être fait de manière manuelle. Ensuite, il faut recompiler deux fois le fichier `LaTeX`.



## 5.3 Quelques règles

Pour conclure, on trouvera ci-dessous quelques règles concernant la constitution d'une bibliographie :

1. De manière générale, une référence doit être incluse dans une bibliographie seulement si elle est utilisée dans le texte. Comme expliqué précédemment, même si le fichier `BIB` contient des références non cités dans le document qui l'utilise, seules les références citées ne figureront dans la bibliographie produite par `bibtex`. Néanmoins, pour rajouter des références non citées, il faut utiliser la commande `\nocite` en précisant les clefs des références non citées à inclure. Si on souhaite inclure l'intégralité de la liste bibliographique (sans tenir compte du fait que les références soient citées ou non), il suffit d'utiliser la commande `\nocite` avec `*` comme unique argument.
2. Le titre d'un ouvrage est écrit en italique. Le titre d'une revue est écrit en italique. A noter que le formatage est automatique si on utilise `bibtex`.

CHAPITRE 6

## Diaporamas et présentations avec $\LaTeX$

Il existe plusieurs classes de document  $\LaTeX$  pouvant être utilisées pour produire un diaporama. Les classes `seminar` et `slides` sont deux des plus anciennes classes disponibles à cet effet. Ici, nous proposons l'utilisation d'une autre classe : `beamer`. Cette classe permet de créer assez facilement des diaporamas dynamiques, un peu comme ce qui peut être fait avec des logiciels comme PowerPoint, mais en moins lourd. A noter qu'il existe une autre classe à peu près similaire, `prosper`.

### 6.1 Faire des diaporamas avec `beamer`

Contrairement aux autres classes déjà étudiées (la classe `article`, par exemple), avec la classe `beamer`, il faut placer les commandes `\title`, `\author` et `\date` dans le préambule.

Pour créer un transparent (i.e. une page), il faut utiliser la commande `\frame`. Le texte qui y sera encapsulé constituera ainsi un transparent. On peut rajouter un titre au transparent à l'aide de la commande `\frametitle` qui prend évidemment en argument le titre à donner.

Dans le but de créer une présentation un peu dynamique, on peut utiliser la commande `\pause` dans un transparent. Il existe d'autres manière de procéder en particulier si on utilise un environnement tel que `itemize`. Pour plus de détails, consulter l'aide sur le site de `beamer`.

Le document peut être enrichi, comme tout document  $\LaTeX$ , d'une structure hiérarchique. Pour cela, il faut utiliser les commandes standards de sectionnement. Au final, cela donne par exemple :

```
\frame{\titlepage}

\section[Plan de l'exposé]{}
\frame{\tableofcontents}

\section{Introduction}

\frame{
\frametitle{Contexte}
blah blah...
}

\frame{
\frametitle{Problème étudié}
blah blah...
\pause
et blah blah...
}
```

Enfin, il est à noter qu'il existe plusieurs thèmes possibles, i.e. plusieurs apparences possibles pour un diaporama. Pour choisir un thème, il faut utiliser la commande `\usetheme` dans le préam-



bule. Par exemple, si on souhaite utiliser le thème `Singapore`, il faut utiliser la commande ainsi : `\usetheme{Singapore}`. Pour connaître les différents thèmes existants, consulter l'aide de `beamer`. Voici ci-dessous un exemple complet (en anglais) sur le site officiel de `beamer` :

```
\documentclass{beamer}

\usepackage{beamerthemesplit}

\title{Example Presentation Created with the Beamer Package}
\author{Till Tantau}
\date{\today}

\begin{document}

\frame{\titlepage}

\section[Outline]{}
\frame{\tableofcontents}

\section{Introduction}
\subsection{Overview of the Beamer Class}
\frame
{
  \frametitle{Features of the Beamer Class}

  \begin{itemize}
    \item<1-> Normal LaTeX class.
    \item<2-> Easy overlays.
    \item<3-> No external programs needed.
  \end{itemize}
}
\end{document}
```

## 6.2 Quelques recommandations

Voici quelques recommandations pour la conception d'un diaporama :

- Il faut éviter de mettre trop de texte par page. Se limiter, par exemple, à moins de dix lignes par page. Il y a deux raisons à cela : d'abord dans un soucis de clarté et surtout pour ne pas être tenté de lire ce qui y est écrit.
- Il faut éviter les animations trop chargées. C'est un grand risque si on utilise des logiciels comme PowerPoint qui est moindre si on utilise  $\LaTeX$  (les fonctionnalités étant moindres).

## 6.3 Exercice

**Exercice 15** Utiliser la classe `beamer` pour faire un diaporama pour présenter votre curriculum vitae (cursus scolaire, stage, ...).

Dans cette dernière partie, nous proposons quelques compléments utiles à la production de documents écrits à l'aide de  $\LaTeX$ .

## 7.1 Générer une table des matières

Il suffit d'inclure la commande `\tableofcontents` à l'endroit où l'on souhaite inclure la table des matières (en général, soit au début soit à la fin d'un document). Pour rajouter la table des matières dans la table des matières, il faut rajouter la commande `\addcontentsline{toc}{section}{Table des matières}` si celle-ci est au niveau `section` (à adapter si besoin est). Cela donne par exemple :

```
\addcontentsline{toc}{section}{Table des matières}
\tableofcontents
```

## 7.2 Générer un index

Il faut d'abord faire appel au package `makeidx`. Ensuite, pour créer un index, il suffit de rajouter devant le nom que l'on souhaite faire figurer en index la commande `\index{texte}` où `texte` est le texte qui sera ajouter en index. Pour placer l'index à l'endroit souhaité, il suffit d'utiliser l'instruction `\printindex`. L'index n'apparaîtra pas dans la table des matières, il faut donc la rajouter à la main à l'aide de la commande `\addcontentsline{toc}{section}{Index}` (cf. la partie précédente). Ensuite il faut compiler le fichier, puis compiler l'index au moyen de la commande (sur le terminal) `makeindex MonFichier` et ensuite compiler à nouveau le fichier  $\LaTeX$ .

Il est possible de créer des sous-entrées en utilisant la commande `\index` de la manière suivante : `\index{clef primaire!clef secondaire}`.

## 7.3 Créer un PDF navigable

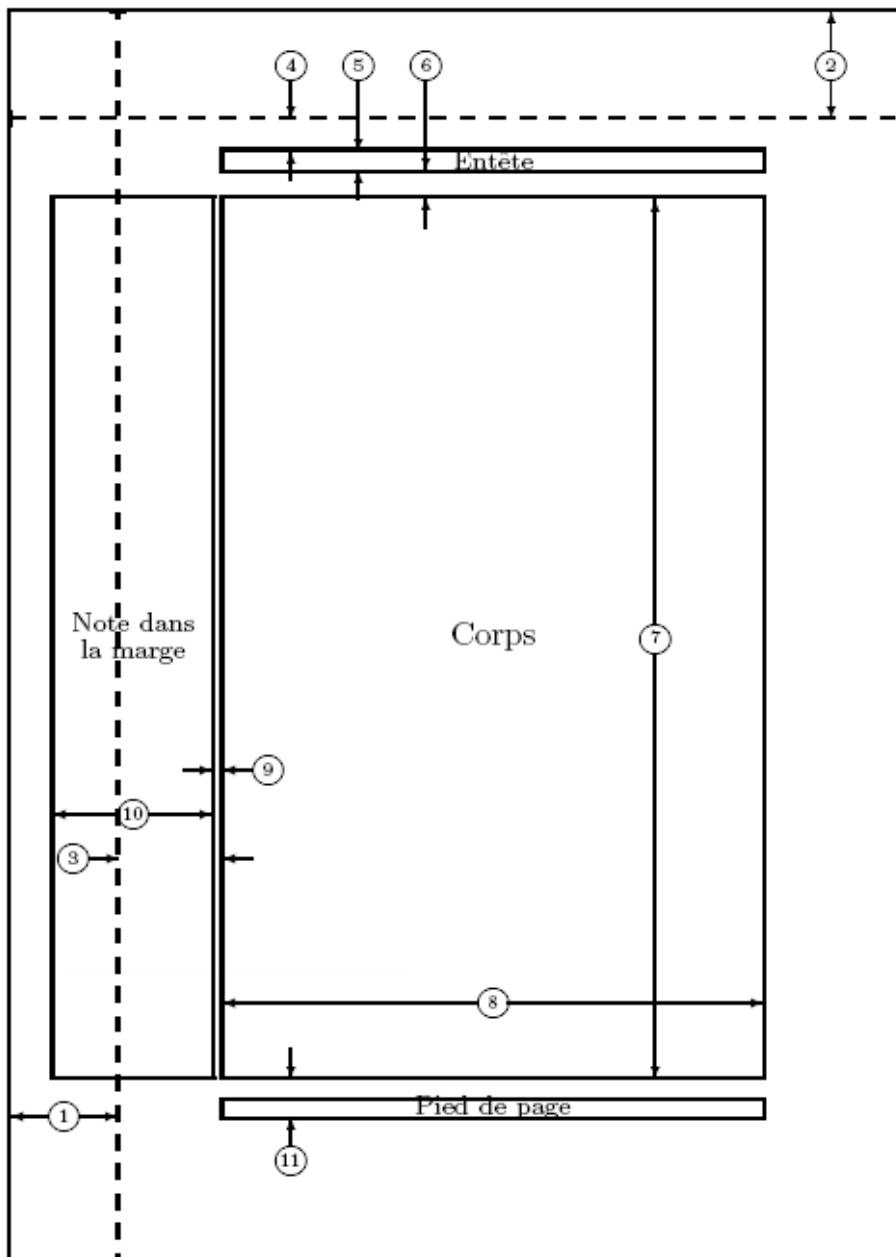
En incluant la package `hyperref`, il est possible de créer un fichier PDF navigable à partir d'un fichier  $\LaTeX$ . Des liens (cliquables) sont créés au niveau de la table des matières, de la bibliographie, de l'index et des équations. Dans ce cas, il faut impérativement utiliser la commande `dvipdf` (ou `dvipdfm`) pour créer le fichier PDF à partir du fichier DVI. Ainsi, on se trouve heurté à un conflit si on souhaite utiliser utiliser le package `psricks` qui nécessite de passer par la commande `dvips`.

## 7.4 Gérer les dimensions du document

Il est possible de modifier les définitions des bordures, marges, etc. à l'aide de commandes placées dans le préambule. Par exemple, pour produire ce document, les commandes suivantes ont permis de le redimensionner comme vous le voyez :

```
\topmargin      0mm
\oddsidemargin  0mm
\evensidemargin 0mm
\hoffset       0mm
\voffset       10mm
\textheight    243.5mm
\textwidth     160.0mm
\columnsep    4.1mm
\parindent     1.0em
\headsep       0mm
\headheight    0pt
\lineskip      0pt
\normallineskip 0pt
\footskip      36pt
```

La figure ci-dessous explique les différents éléments pour le redimensionnement (extrait du polycopié *Une courte (?) introduction à  $\LaTeX 2_{\epsilon}$* ) :



1	un pouce + <code>\hoffset</code>	2	un pouce + <code>\voffset</code>
3	<code>\evensidemargin = 70pt</code>	4	<code>\topmargin = 22pt</code>
5	<code>\headheight = 13pt</code>	6	<code>\headsep = 19pt</code>
7	<code>\textheight = 595pt</code>	8	<code>\textwidth = 360pt</code>
9	<code>\marginparsep = 7pt</code>	10	<code>\marginparwidth = 106pt</code>
11	<code>\footskip = 27pt</code>		<code>\marginparpush = 5pt</code> (non affiché)
	<code>\hoffset = 0pt</code>		<code>\voffset = 0pt</code>
	<code>\paperwidth = 597pt</code>		<code>\paperheight = 845pt</code>

## 7.5 Ecrire une lettre avec $\LaTeX$

Pour écrire un lettre avec  $\LaTeX$ , il faut utiliser la classe `letter`. Les différents éléments de la lettre seront placés aux endroits conventionnels selon le langage déclaré lors de l'appel au package `babel`.

Une partie des éléments de la lettre figure dans le préambule : il s'agit essentiellement des commandes `\name{}`, `\address{}` et `\signature{}`.

Le reste de la lettre figure ensuite dans le corps principal du fichier. Il faut utiliser l'environnement `\letter`. Les mots d'introduction sont placés automatiquement à l'aide de la commande `\opening` et la formule de politesse à l'aide de la commande `\closing`.

## 7.6 Exercices

**Exercice 16** Reprendre le texte extrait du livre de René Cagnat et choisir plusieurs mots qui seront utilisés pour un index. Produire un fichier PDF navigable contenant une table des matières.

**Exercice 17** Ecrire une lettre fictive de candidature pour un emploi d'ingénieur mathématicien dans une entreprise ou un laboratoire de recherche.

### CHAPITRE 8

## Ressources électroniques

Pour répondre à une question de base, il y a Foire Aux Questions sur  $\LaTeX$ , mais qui n'est plus mis à jour depuis quelques années. Autrement, en partant du site web CTAN (Comprehensive  $\TeX$  Archive Network), il est possible de télécharger différents documents d'introduction à  $\LaTeX$ , en français ou en anglais. Parmi ceux-ci, on peut citer les suivants :

- Marc Baudoin 'Babafou', *Apprends  $\LaTeX$*  (ENSTA), 1994-1997.
- Benjamin Bayart, *Joli manuel pour  $\LaTeX$  2 $\epsilon$  (guide local de l'ESIEE)*, 1995.
- Michael Downes, *Short math guides for  $\LaTeX$* , 2000
- Tetuso Iwakuma and Tetsuo Furukawa, *A guide to use macros and style files in  $\LaTeX$* , 1994.
- Tobias Oetiker, Hubert Partl, Irene Hyna et Elisabeth Schlegl (traduction française par Matthieu Herrb), *Une courte (?) introduction à  $\LaTeX$  2 $\epsilon$  (ou  $\LaTeX$  2 $\epsilon$  en 88 minutes)*, 1999.
- Vincent Séguin, *Aide-mémoire  $\LaTeX$* , 2000.

Ces documents sont quelque peu anciens mais permettent de pratiquer un minimum  $\LaTeX$ . En conséquence, certaines commandes peuvent être devenues obsolètes suite aux évolutions constantes de  $\LaTeX$ .

### CHAPITRE 9

## Annexe : règles typographiques de la langue française

La typographie d'une langue précise l'ensemble des règles de composition d'un texte dans cette langue. En particulier, l'usage de la ponctuation répond des règles bien précises. En français, les symboles de ponctuation constitués de deux parties (: ; ? !) sont séparés du texte d'un espace avant et après, alors que les autres symboles (. , ( ) -) ne sont séparés du texte que d'un espace après (et donc ils sont collés au mot d'avant). En revanche, en anglais, toutes les ponctuations du texte sont séparés d'un espace après (ils sont donc toujours collés au mot d'avant).