

INTRODUCTION À L^AT_EX

C. CHARLES*, L. LECHARLIER[†] et F. RENAUD[‡]

RÉSUMÉ

Cette note technique est une initiation au traitement de texte L^AT_EX. L'objectif visé est de pouvoir rédiger un document scientifique tel qu'un travail de fin d'études en L^AT_EX. Ainsi, après la lecture de cette introduction, le lecteur sera capable de concevoir un document, d'effectuer sa mise en page, d'y inclure des images et des tableaux. D'autres notions plus avancées sont également vues telles que créer une table des matières, un index ou une bibliographie.

SUMMARY

This note aims to introduce the L^AT_EX word processing. The objective is to be able to compose a final report by use of L^AT_EX. Therefore, you will be able to produce documentation, to design document and to include tables and figures. Advanced notions as generation of bibliography, index or table of contents is also explained.

1. INTRODUCTION

1.1. Historique

Vu ses attentes en matière d'édition d'œuvres, Donald KNUTH, universitaire américain (informaticien et mathématicien), créa en 1977-1978 T_EX, un puissant programme de mise en forme de documents permettant de réaliser aisément des formules mathématiques complexes [KNUTH, 1986]. Sa philosophie était basée sur les principes suivants :

*Chargée de cours à la Faculté universitaire des Sciences agronomiques de Gembloux (Unité de Statistique, Informatique et Mathématique appliquées)

[†]Assistant à la Faculté universitaire des Sciences agronomiques de Gembloux (Unité de Statistique, Informatique et Mathématique appliquées)

[‡]Assistant à la Faculté universitaire des Sciences agronomiques de Gembloux (Unité de Statistique, Informatique et Mathématique appliquées) de 2004 à 2007.

au niveau de la qualité : « Nous ne voulions pas produire de bons documents, nous voulions produire les meilleurs. »,

au niveau de l'archivage : « Mon but était de m'y prendre de telle manière que, si l'on sauvegardait les spécifications d'un livre, nos descendants seraient encore capables d'éditer le même livre en 2086. ».

Entre 1984 et 1990, Leslie LAMPORT créa une version spéciale de $\text{T}_{\text{E}}\text{X}$, nommée $\text{\LaTeX}$, afin que l'utilisateur puisse profiter de la puissance de $\text{T}_{\text{E}}\text{X}$ avec une plus grande facilité [LAMPORT, 1994].

1.2. Caractéristiques

$\text{\LaTeX}$ ("Latek") est avant tout un système performant de préparation de documents à la fois libre et gratuit.

$\text{\LaTeX}$ fait partie des logiciels de traitement de texte non WYSIWYG ("*What You See Is What You Get*"), ce qui veut dire que le document est décrit par un ensemble de commandes dans un fichier source et que le résultat ne peut être visualisé qu'après une compilation. En fait, $\text{\LaTeX}$ est principalement basé sur le fond et non sur la forme. L'idée est de laisser, d'une part, l'auteur rédiger le document et, d'autre part, le styliste le mettre en page. Ainsi, cette mise en page est automatique mais bien sûr paramétrable.

Ce traitement de texte est très puissant ; il dispose d'une typographie professionnelle. Son succès provient entre autres de la facilité avec laquelle n'importe quelle formule mathématique peut être introduite dans le document, ainsi que de l'excellente qualité visuelle de ces expressions mathématiques.

$\text{\LaTeX}$ a l'avantage de fonctionner sur tout type de plate-forme (Unix, Windows, Sun, etc.) et de pouvoir être utilisé, copié et distribué librement. Sa compatibilité est aussi bien ascendante que descendante.

Enfin, la taille d'un fichier $\text{\LaTeX}$ est 1000 fois plus petite que celle utilisée pour un fichier de type Word.

Ainsi, le succès de $\text{\LaTeX}$ est indéniable, aussi bien dans les sites universitaires que dans les sites commerciaux et partout dans le monde. Les deux handicaps de ce traitement de texte concernent, d'une part, l'insertion d'image qui, comme pour Word, demande souvent le recours à des outils spécialement dédiés au dessin, et, d'autre part, le temps d'apprentissage nécessaire à la maîtrise de ce logiciel.

1.3. Plan

Après cette introduction, le paragraphe 2 présente les grandes étapes de la création d'un document. Le paragraphe 3 explique la structure générale et les commandes de base se trouvant dans le fichier source. Le paragraphe 4 s'intéresse aux environnements particuliers, tels que les listes, les formules mathématiques, etc. Le paragraphe 5 précise la manière dont il faut procéder pour inclure dans

le document des tableaux, des images, des références ou des annotations. Le paragraphe 6 s'attache aux tables des matières, index et bibliographie. Enfin, nous présentons quelques notions qui permettront à ceux qui le désirent de persévérer dans la compréhension de L^AT_EX (paragraphe 7).

2. INSTALLATION ET CRÉATION D'UN DOCUMENT

2.1. Installation

Comme précisé dans l'introduction, L^AT_EX fait partie des logiciels de traitement de texte non WYSIWYG : le fichier source contenant un ensemble de commandes doit être compilé pour donner le document attendu. Pour le fichier source, un simple éditeur de texte peut être utilisé. Le compilateur MikTeX est suggéré et peut être téléchargé à l'adresse <http://www.miktex.org/>. Cependant, il existe aussi des éditeurs facilitant la tâche avec des interfaces simples permettant la compilation, la visualisation, l'impression, etc. TeXnicCenter en est un exemple et peut être téléchargé à l'adresse <http://www.toolscenter.org>.

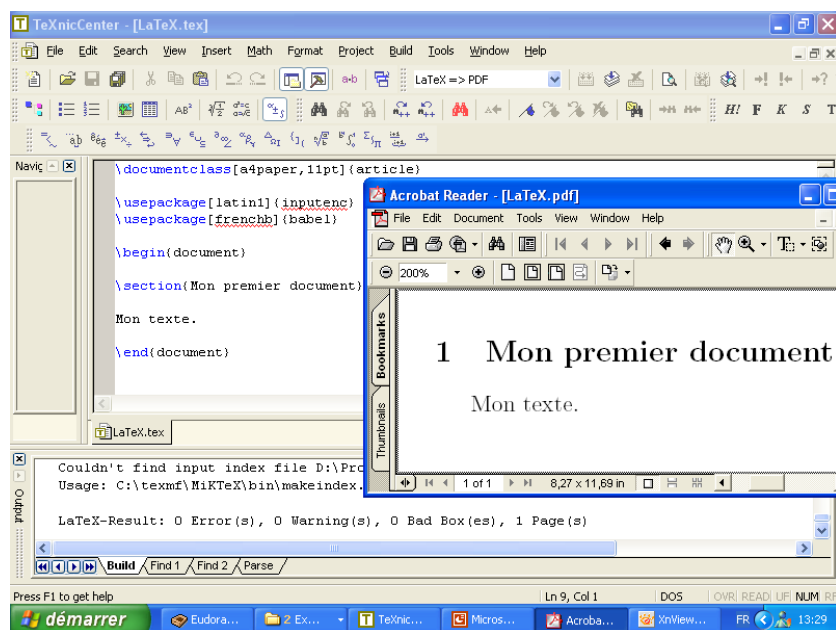


Figure 1. Premier document.

2.2. Étapes de création d'un document

Il faut d'abord créer un fichier source $\text{\LaTeX}$ contenant toutes les commandes nécessaires à la description du document. En réalité, le texte source est encodé à l'aide d'un éditeur de texte et sauvegardé dans un fichier dont l'extension est ".tex". Ensuite, ce fichier est compilé, ce qui le traduit en un fichier servant de base à d'autres programmes spécialisés dans l'affichage ou l'impression. Ce nouveau fichier (fichier .dvi) est un fichier *DeVice Independant* (indépendant de tout matériel). Si on veut visualiser le résultat sur écran, on utilise un programme de visualisation approprié au système d'affichage utilisé. Si on veut imprimer le document, il est également nécessaire d'utiliser un programme approprié au type d'imprimante utilisé. Des environnements intégrés existent pour passer rapidement de l'une à l'autre de ces étapes de la réalisation d'un document latex.

Si le fichier source, par exemple monfichier.tex, est encodé à l'aide d'un éditeur de texte quelconque, les commandes pour compiler, visualiser et imprimer le document se font en Ms-dos¹ :

1. compiler le fichier $\text{\LaTeX}$: `latex monfichier.tex`,
2. visualisation du fichier .dvi : `yap monfichier.dvi`,
3. conversion en format PostScript : `dvips monfichier.dvi`,
conversion en format pdf : `dvipdfm monfichier.dvi`.

Il y a la possibilité d'obtenir un fichier .pdf à partir d'un fichier .tex par la commande : `pdflatex monfichier.tex`. Pour obtenir un fichier .pdf à partir du fichier .ps obtenu à la troisième étape, la commande à utiliser est `ps2pdf monfichier.ps`.

2.3. Caractères spéciaux

Le fichier source est la matière brute du document. Il est non seulement composé du texte du document, mais aussi des commandes qui décrivent sa mise en forme. Dès lors, $\text{\LaTeX}$ interprète un certain nombre de caractères comme des commandes. Les principaux caractères spéciaux sont repris dans le tableau 1.

2.4. Espacements

Les espacements dans l'éditeur n'ont aucune influence (exemple : 3 blancs = 1 blanc). Les commandes à utiliser pour effectuer de plus grands espacements entre deux mots sont reprises dans le tableau 2.

Pour les espacements horizontaux et verticaux, il existe des variantes : `\hspace*` et `\vspace*`. Ainsi, `\hspace*{length}` se comporte comme la commande `\hspace`, mais l'espacement produit n'est jamais enlevé, même s'il se trouve en début ou fin de page.

¹commandes valables uniquement sous Windows

Caractères spéciaux	Interprétation
%	début d'un commentaire
\	début d'une commande
{ }	environnement local
~	espace insécable
_	indice
^	exposant
\$	environnement mathématique
&	colonne de tableau
#	début d'une macro

Tableau 1. Caractères spéciaux.

Commandes	Interprétation
\newline ou \\	retour à la ligne
\newpage	saut de page
\hspace{1cm,mm,pt}	espace horizontal de taille 1 cm, mm, pt(point)
\vspace{1cm,mm,pt}	espace vertical de taille 1 cm, mm, pt
ligne vide	changement de paragraphe

Tableau 2. Espacements.

3. STRUCTURE D'UN DOCUMENT $\text{\LaTeX}$

3.1. Structure générale

Tout fichier `.tex` doit obéir à une même structure, telle que présentée ci-après :

```
\documentclass{TypeDeDocument}
% préambule
\begin{document}
% Corps du texte
\end{document}
```

Le préambule sert à placer des définitions supplémentaires dont on aurait besoin dans le corps du texte. Le texte à imprimer doit se trouver dans le corps du texte.

3.2. Type de document

Tout fichier `.tex` commence par la commande :

```
\documentclass[options]{type}
```

qui spécifie le type de document à créer.

Les options sont :

- `10pt`, `11pt`, `12pt` : définition de la taille du caractère,
- `a4paper`, `a5paper` : définition de la taille du papier,
- `landscape` : orientation « paysage »,
- `twocolumn` : texte sur deux colonnes verticales,
- `oneside`, `twoside` : impression recto - recto/verso,
- `openany`, `openright` : emplacement du chapitre (sur n'importe quelle page ou seulement sur une page impaire).

Le type est à choisir selon ce qui suit :

- `article` : création de documents courts (comptes-rendus de travaux ou de réunions),
- `report` : création de documents plus longs (gestion des chapitres),
- `book` : création de livres (similaire à la classe `report`),
- `letter`, `slides`, etc.

L'exemple ci-après indique que nous allons créer un document de quelques pages de format A4 dont la taille du caractère sera de 11pt :

```
\documentclass[11pt,a4paper]{article}
```

D'une manière générale, il faut savoir que, derrière une commande, `[ ]` indique des options tandis que `{ }` indique des informations obligatoires.

3.3. Préambule

Après la commande spécifiant le type de document à créer se trouve le préambule, qui sert à placer des définitions supplémentaires dont on aurait besoin dans le corps du texte. Il permet également de modifier la mise en page et de charger des extensions à $\text{\LaTeX}$.

1. Lorsque nous avons souvent à encoder la même suite de commandes, on peut introduire une nouvelle commande reprenant cette suite de commandes grâce à `\newcommand`. Par exemple,

```
\newcommand{\A}{\mathrm{A}\rightarrow\mathrm{B}}
```

définit la nouvelle commande `\A` réalisant `\mathrm{A}\rightarrow\mathrm{B}`. On a donc créé une commande plus courte à encoder. Taper `\A`, produit le texte suivant : « $A \Rightarrow B$ ».
Un autre exemple est `\newcommand\rit{\hbox{\it I\hspace{-2pt}R}}` qui permet en écrivant `\rit` d'obtenir le symbole $\mathbb{R}$.
2. Il existe différentes commandes permettant de modifier la mise en page. Deux de celles-ci sont reprises dans le tableau 3.

Commandes	Interprétation
<code>\textwidth 122mm</code>	spécification d'une largeur normale du texte d'une page de 122mm
<code>\textheight 202mm</code>	spécification d'une longueur normale du texte d'une page de 202mm

Tableau 3. Mise en page.

Les commandes reprises dans le tableau 4 peuvent aussi être mises dans

Commandes	Interprétation
<code>\title{Titre}</code>	spécification du titre du document
<code>\author{auteur}</code>	spécification du (des) auteur(s) du document
<code>\date{date}</code>	spécification de la date de création du document
<code>\date{}</code>	pas de spécification de date de création du document

Tableau 4. En-tête.

le préambule. Elles contiennent les informations nécessaires pour l'en-tête du document. Il faut alors ajouter la commande :

`\maketitle`

dans le corps du texte pour que toutes ces informations soient affichées.

3. Enfin pour charger des extensions à $\text{\LaTeX}$, la commande principale est :

`\usepackage[options]{extensions}`

Quelques exemples se trouvent dans le tableau 5.

Commandes	Interprétation
<code>\usepackage[latin1]{inputenc}</code>	utilisation des caractères iso8859-1
<code>\usepackage[frenchb]{babel}</code>	adaptation de $\text{\LaTeX}$ au français
<code>\usepackage{amssymb,amsfonts,amsmath}</code>	ajout de symboles

Tableau 5. Extensions.

3.4. Corps du texte

Tout le texte doit se trouver entre les instructions

`\begin{document}`

...

`\end{document}`

C'est donc dans cette partie que peuvent être insérés le texte, les images, les tableaux, etc.

3.5. Division du document

Les classes de document `article`, `report` et `book` offrent à l'utilisateur un certain nombre de commandes (tableau 6) qui automatisent et facilitent la composition de document.

Division	Classe	Commande
Partie	<code>article</code> , <code>report</code> , <code>book</code>	<code>\part</code>
Chapitre	<code>report</code> , <code>book</code>	<code>\chapter</code>
Section	<code>article</code> , <code>report</code> , <code>book</code>	<code>\section</code>
Sous-section	<code>article</code> , <code>report</code> , <code>book</code>	<code>\subsection</code>
Sous-sous-section	<code>article</code> , <code>report</code> , <code>book</code>	<code>\subsubsection</code>

Tableau 6. Divisions d'un document

Par exemple, la commande `\chapter{Titre}` va disposer elle-même `Titre` comme titre du chapitre. La numérotation du chapitre se fait alors automatiquement. La taille et le style de `Titre` sont également automatiques. La commande `\chapter*{Titre}` empêche la numérotation de ce chapitre. Signalons qu'en général on ne numérote pas le chapitre consacré à l'introduction. Enfin, la commande `\chapter[Abr]{Titre}` prendra `Abr` à la place de `Titre` dans la table des matières. Ceci est souvent utilisé pour des titres trop longs.

4. ENVIRONNEMENTS

Les dispositions du texte (centré, gras, etc) et le mode mathématique sont réalisés en utilisant la notion d'environnement. Un environnement est spécifié par des commandes qui permettent d'entrer ou de sortir de la zone où le texte est traité de manière particulière :

```
\begin{environnement}  
...  
\end{environnement}
```

sachant que le mot `environnement` est le nom de l'environnement souhaité.

Par exemple l'environnement `verbatim` permet de faire apparaître un texte tel qu'il est écrit, c'est-à-dire que le texte mis dans l'environnement `verbatim` ne sera pas interprété.

Code 1

```
La commande pour l'environnement flushright est  
\begin{verbatim}  
\begin{flushright}  
Le texte  
\end{flushright}  
\end{verbatim}
```


Résultat 1

La commande pour l'environnement `flushright` est

```
\begin{flushright}
Le texte
\end{flushright}
```

On voit donc que les commandes d'ouverture et de fermeture de l'environnement `flushright` (paragraphe 4.1.) ne sont pas interprétées mais écrites intégralement dans le texte. Enfin, on remarque que l'environnement `verbatim` écrit le texte en caractère de type `teletype` (paragraphe 4.3.).

4.1. Environnements d'alignement

Si on veut changer le type d'alignement du texte, il existe 3 environnements :

- l'environnement `flushleft` pour aligner le texte à gauche,
- l'environnement `flushright` pour aligner le texte à droite,
- l'environnement `center` pour centrer le texte.

Illustrons ce qui précède par quelques exemples.

Code 2

```
\begin{flushleft}
Mon texte est aligné à gauche.
\end{flushleft}

\begin{flushright}
Mon texte est aligné à droite.
\end{flushright}

\begin{center}
Mon texte est centré.
\end{center}
```

Résultat 2

Mon texte est aligné à gauche.

Mon texte est aligné à droite.

Mon texte est centré.

Notons que $\text{\LaTeX}$ effectue lui-même la mise en page et les coupures de lignes.

4.2. Environnements de listes

1. Pour les listes simples, l'environnement est `itemize`.

Code 3

```
Les principales méthodes d'intégration sont :
\begin{itemize}
  \item l'intégration par substitution,
  \item l'intégration par changement de variable,
  \item[---] l'intégration par parties.
\end{itemize}
```

Résultat 3

Les principales méthodes d'intégration sont :

- l'intégration par substitution,
- l'intégration par changement de variable,
- l'intégration par parties.

On peut remarquer que le code `[--]` qui suit le troisième `\item` a pour effet de créer un tiret plus long.

2. Pour les listes de description, l'environnement est `description`.

Code 4

```
Les différents types de coniques et leur excentricité  $e$  sont :
\begin{description}
  \item[l'ellipse :]  $0 < e < 1$ ,
  \item[la parabole :]  $e = 1$ ,
  \item[l'hyperbole :]  $e > 1$ .
\end{description}
```

Résultat 4

Les différents types de coniques et leur l'excentricité e sont :

l'ellipse : $0 < e < 1$,

la parabole : $e = 1$,

l'hyperbole : $e > 1$.

Notons que `[ ]` est à option.

3. Pour les listes d'énumérations, l'environnement est `enumerate`.

Code 5

```
Les différentes étapes de l'étude d'une fonction  $f$  sont :
\begin{enumerate}
  \item déterminer le domaine et les racines de  $f$ ,
  \item calculer  $f'$  et en déduire les extrêmes éventuels de  $f$ ,
  \item calculer  $f''$  et en déduire les inflexions éventuelles de  $f$ ,
  \item dessiner le graphe de  $f$ .
\end{enumerate}
```

Résultat 5

Les différentes étapes de l'étude d'une fonction f sont :

1. déterminer le domaine et les racines de f ,
2. calculer f' et en déduire les extrêmes éventuels de f ,
3. calculer f'' et en déduire les inflexions éventuelles de f ,
4. dessiner le graphe de f .

Cependant, il faut remarquer que le type d'énumération donné par l'environnement `enumerate` dépend de son niveau d'imbrication dans d'autres environnements `enumerate` (maximum quatre niveaux). Les différents types sont décrits dans le tableau 7.

Niveau	Type d'énumération
0	1., 2., 3., ...
1	(a), (b), (c), ...
2	i., ii., iii., ... (chiffre romain minuscule)
3	A., B., C.

Tableau 7. Types d'énumération en fonction du niveau d'imbrication

Si on veut pouvoir choisir le type d'énumération, il faut alors utiliser le package `enumerate` (ne pas oublier d'ajouter `\usepackage{enumerate}` dans le préambule). Pour choisir le type d'énumération, on utilise la commande : `\begin{enumerate}[type d'énumération]`, avec comme type d'énumération, un de ceux se trouvant dans le tableau 8.

Type	Résultat
1	1, 2, 3, ...
a	a, b, c, ...
A	A, B, C
i	i, ii, iii, iv, ... (chiffres romains minuscules)
I	I, II, III, IV, ... (chiffres romains majuscules)

Tableau 8. Types d'énumération

Par exemple, si on veut faire une énumération du type Cas 1, Cas 2, ..., on va utiliser la commande `\begin{enumerate}[C{a}s 1]`.
 Il faut remarquer que le caractère a du mot cas a été mis entre accolades ; sinon, il aurait aussi été énuméré puisqu'il définit un type d'énumération.

4.3. Environnements de texte

Pour changer la taille ou le type de caractère, on utilise également la notion d'environnement. La syntaxe utilisée peut être `\commande{Texte}` ou `{\commande Texte}`.

1. Les environnement pour changer la police des caractères sont décrits dans le tableau 9.

Commandes	Résultats
<code>\textit{texte en italique}</code>	<i>texte en italique</i>
<code>\textbf{texte en gras}</code>	texte en gras
<code>\texttt{texte en caractères Tele Type}</code>	texte en caractères Tele Type
<code>\textsc{texte en petites capitales}</code>	TEXTE EN PETITES CAPITALES
<code>\textrm{texte en caractères romains}</code>	texte en caractères romains
<code>\textsf{texte en caractères linéaux}</code>	texte en caractères linéaux
<code>\emph{texte en style emphatique}</code>	<i>texte en style emphatique.</i>

Tableau 9. Environnements pour changer la police des caractères.

Pour utiliser la deuxième syntaxe permettant de changer la police des caractères, les commandes à utiliser sont celles du tableau ci-dessus moins `text`. Par exemple pour mettre le texte en italique, la commande est `{\it le texte}` qui donnera comme résultat *le texte*.

2. Les environnements pour changer la taille des caractères sont par ordre croissant :
`\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`,
`\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`

Code 6

```
{\tiny La taille des caractères} {\small de mon texte est} {\large de plus en plus} {\huge grande.}
```

Résultat 6

La taille des caractères de mon texte est de plus en plus grande.

Il y a aussi la possibilité de combiner les commandes.

Code 7

```
{\small \textsf{Le rang d'une matrice est égale au rang de sa transposée.}}
```

Résultat 7

Le rang d'une matrice est égale au rang de sa transposée.

3. La commande pour encadrer du texte est `\fbox{Mon texte}` qui donne comme résultat Mon texte.
4. La commande pour souligner du texte est `\underline{Mon texte}` qui donne comme résultat Mon texte.

4.4. L'environnement mathématique

L^AT_EX a été conçu à la base pour faciliter l'encodage de formules mathématiques complexes comme par exemple la formule de la variance :

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Si on veut écrire une formule dans le texte, on utilise `$ Formule $` comme par exemple `$y= \sin x$` qui donne comme résultat $y = \sin x$. Par contre, si on veut mettre la formule en évidence hors du texte, on utilise `\[ Formule \]`. C'est l'environnement utilisé dans l'exemple de la variance ci-avant.

Si on veut numéroter les formules, l'environnement utilisé est

```
\begin{equation}
  Formule
\end{equation}
```

Le même environnement avec le symbole `*` supprime la numérotation :

```
\begin{equation*}
  Formule
\end{equation*}
```

Lorsqu'on se trouve dans un environnement mathématique, les lettres sont en italique et les espaces sont tous ignorés. Si on veut ajouter du texte, des espaces ou encadrements, les commandes à utiliser sont :

Texte	:	<code>\textrm{texte}</code>
Espace	:	<code>\, , \; \quad \qquad</code>
Encadrement de la formule	:	<code>\boxed{formule}</code>

Les principaux symboles mathématiques sont repris dans le tableau 10. On constate que les indices et exposants se notent avec deux commandes simples `_{xxx}` et `^{xxx}`.

$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\int$	<code>\int</code>
$\prod$	<code>\prod</code>	$\sum_{i=1}^n$	<code>\sum_{i=1}^n</code>	$\pm$	<code>\pm</code>
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>	$\leq$	<code>\leq</code>
$\geq$	<code>\geq</code>	$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni</code>	$\approx$	<code>\approx</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Rightarrow$	<code>\Rightarrow</code>	$\iff$	<code>\iff</code>
$\cos$	<code>\cos</code>	$\sin$	<code>\sin</code>	$\ln$	<code>\ln</code>
$\vec{a}$	<code>\vec{a}</code>	$\grave{a}$	<code>\grave{a}</code>	$\acute{a}$	<code>\acute{a}</code>
$\tilde{a}$	<code>\tilde{a}</code>	$\bar{a}$	<code>\bar{a}</code>	$\hat{a}$	<code>\hat{a}</code>
$\langle$	<code>\langle</code>	$\rangle$	<code>\rangle</code>	$\overbrace{aaa}^b$	<code>\overbrace{aaa}^b</code>
$\underbrace{aaa}_b$	<code>\underbrace{aaa}_b</code>	$\dots$	<code>\dots</code>	$\vdots$	<code>\vdots</code>
$\frac{2}{3}$	<code>\frac{2}{3}</code>	$\sqrt[3]{x}$	<code>\sqrt[3]{x}</code>	$\lim_{x \rightarrow \infty} f(x)$	<code>\lim_{x \rightarrow \infty} f(x)</code>

Tableau 10. Principaux symboles mathématiques

Lorsqu'on a un tableau de formules ou une formule qui s'étend sur plusieurs lignes, l'environnement utilisé est, avec numérotation :

```
\begin{eqnarray}
  Formule\\
  Formule
\end{eqnarray}
```

sans numérotation :

```
\begin{eqnarray*}
  Formule\\
  Formule
\end{eqnarray*}
```

De plus, l'environnement `eqnarray` est aussi un environnement mathématique. Par conséquent, contrairement à l'environnement `array` (paragraphe 5.1.), il ne faut pas passer dans un environnement mathématique pour pouvoir l'utiliser. L'environnement `eqnarray` permet aussi d'aligner les formules sur un symbole, pour cela il faut entourer ce symbole de deux symboles `&`.

Code 8

```
\begin{eqnarray*}
f(x)&=& x^3+2x^2-3-(x^2+2x) \\\
&=& x^3+2x^2-3-x^2-2x \\\
&=& x^3+x^2-2x-3
\end{eqnarray*}
```

Résultat 8

$$\begin{aligned} f(x) &= x^3 + 2x^2 - 3 - (x^2 + 2x) \\ &= x^3 + 2x^2 - 3 - x^2 - 2x \\ &= x^3 + x^2 - 2x - 3 \end{aligned}$$

Dans une expression mathématique, on utilise très souvent des délimiteurs tels que (, [ou { pour une meilleure lisibilité. Pour augmenter encore cette lisibilité $\LaTeX$ possède des délimiteurs adaptatifs c'est-à-dire des délimiteurs dont la taille va s'adapter à la taille de l'expression à délimiter. La syntaxe est `\leftDélimateur Expression \rightDélimateur`. De plus si on ne veut pas de délimiteur d'un côté de l'expression alors le délimiteur à coller à `\left` ou `\right` est «·». Par exemple `\left(\frac{x+1}{2}\right)` va donner $(\frac{x+1}{2})$.

5. TABLEAUX, FIGURES, RÉFÉRENCES ET ANNOTATIONS

5.1. Les tableaux

Pour insérer un tableau dans le texte, la syntaxe à suivre est :

```
\begin{tabular}[position]{description}
...
\end{tabular}
```

Pour insérer un tableau dans un environnement mathématique, on peut suivre la syntaxe suivante :

```
\begin{array}[position]{description}
...
\end{array}
```

Dans la partie **description**, il faut utiliser les symboles suivants :

r	texte aligné à droite,
l	texte aligné à gauche,
c	texte centré,
p{1}	largeur fixe de 1 cms,
 	ligne verticale.

Si on choisit un alignement du type **r**, **l**, ou **c**, le texte d'une cellule sera mis sur une seule ligne. Si le texte est trop long pour la largeur du tableau ou de la page, le texte sortira du tableau et/ou de la page. Par contre si on choisit l'option **p{1}**, le texte sera aligné à gauche ; si le texte est plus long que largeur **1**, alors le texte sera coupé et mis sur le nombre de lignes nécessaire mais toujours dans la même cellule du tableau comme dans l'exemple ci-après.

Code 9

```
\begin{tabular}{|l|p{5cm}|} \hline
Texte court : & Voici un texte court \\ \hline
Test long : & ce texte-ci est trop long et sera donc mis sur
plusieurs lignes \\ \hline
\end{tabular}
```

Résultat 9

Texte court :	Voici un texte court.
Test long :	Ce texte-ci est trop long et sera donc mis sur plusieurs lignes.

Il existe deux largeurs prédéfinies intéressantes : `\textwidth` pour que le tableau possède la largeur du texte et `\linewidth` pour que le tableau possède la largeur d'une ligne de texte.

Les **positions** possibles sont :

- **t** : extrémité supérieure du tableau doit être positionnée au même niveau que le texte entourant ce tableau,
- **b** : inférieure,
- **]** : milieu.

Les commandes utiles sont

- **&** permet de passer d'une colonne à une autre colonne sur la même ligne,
- **** permet de passer à la ligne suivante,
- **\hline** trace une ligne horizontale sur toute la largeur du tableau,
- **\cline{x-y}** trace une ligne horizontale de la colonne **x** à la colonne **y**,
- **\multicolumn{nbcol}{align}{Txt}** écrit **Txt** sur une largeur **nbcol** avec le format **align**,
- **\multirow{nblign}{taille}{Txt}** écrit **Txt** sur une longueur **nblign** avec le format **taille**. Pour pouvoir utiliser cette commande, il faut ajouter le package **multirow** dans le préambule du document (`\usepackage{multirow}`).

Ci-dessous se trouvent trois exemples différents d'utilisation de tableaux.

Code 10

```
\begin{tabular}{|c|c|c|c|} \hline
A& B& A ou B & A et B \\ \hline
Vrai & Vrai & Vrai & Vrai \\
Vrai & Faux & Vrai & Faux \\
Faux & Vrai & Vrai & Faux \\
Faux & Faux & Faux & Faux \\ \hline
\end{tabular}
```


Résultat 10

A	B	A ou B	A et B
Vrai	Vrai	Vrai	Vrai
Vrai	Faux	Vrai	Faux
Faux	Vrai	Vrai	Faux
Faux	Faux	Faux	Faux

On peut également réaliser des tableaux dans un environnement mathématique. Par exemple, une matrice ou un système de deux équations à deux inconnues ne sont rien d'autre que des tableaux avec des délimiteurs adaptatifs.

Code 11

```
\[
\left(
\begin{array}{ccc}
x_{11} & \cdots & x_{1p} \\
\vdots & \ddots & \vdots \\
x_{n1} & \cdots & x_{np}
\end{array}
\right)
```

Résultat 11

$$\left(\begin{array}{ccc} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{array} \right)$$

Code 12

```
\[
\left\{
\begin{array}{l}
2e^x + \log y = 1 \\
\sin 2x + y^4 = 0
\end{array}
\right.
```

Résultat 12

$$\left\{ \begin{array}{l} 2e^x + \log y = 1 \\ \sin 2x + y^4 = 0 \end{array} \right.$$

5.2. Les figures et les images

Les possibilités graphiques de $\text{\LaTeX}$ ont des limites. Dès lors, pour insérer des graphiques, on réalise généralement ceux-ci dans un outil indépendant de $\text{\LaTeX}$ et on les incorpore par la suite dans le document. A ce moment, on a besoin d'une extension à $\text{\LaTeX}$ comme `graphicx`, `epsfig`, etc, pour pouvoir inclure des images de différents formats comme `.eps`, `.png`, `.jpeg`. De plus il faut mettre une option particulière et utiliser une commande de compilation différente selon le type d'image à inclure. Le tableau 11 en fait un récapitulatif.

Type d'image	Option	Compilation
<code>.ps</code> , <code>.eps</code>	<code>\usepackage{graphicx}</code>	latex suivi de <code>dvips</code>
	<code>\usepackage{graphicx}</code>	latex suivi de <code>dvipdfm</code>
<code>.jpg</code> , <code>.png</code>	<code>\usepackage{graphicx}</code>	<code>pdflatex</code>

Tableau 11. Commandes de compilation et packages pour l'inclusion d'image

La syntaxe à utiliser à l'endroit où on veut insérer l'image est :

```
\includegraphics[Options]{NomImage}
```

Les options possibles sont :

- `angle` : rotation,
- `height` : hauteur,
- `width` : largeur,
- `scale` : facteur d'échelle.

Exemple :

```
\includegraphics[width=5cm,angle=90]{normal.jpg}
```

5.3. Les objets flottants

Les objets flottants sont des objets composés de tableaux, textes, graphiques, etc., dont la position d'impression dans le document est laissée au moteur de mise en forme. Les environnements `figure` et `table` permettent d'inclure dans un document des figures et des tables flottantes. Par exemple, il est possible d'insérer un environnement `tabular` à l'intérieur de l'un de ces environnements. Les numérotations des figures et des tables se font indépendamment l'une de l'autre.

L'appel de ces environnements se fait sous la forme :

```
\begin{figure}[pos] ... \end{figure}
\begin{table}[pos] ... \end{table}
```

où `[pos]` indique le positionnement :

- `[h]` : dans le texte à l'endroit appelé,
- `[!h]` : ici absolument,
- `[t,b]` : respectivement en haut et en bas de page,
- `[p]` : sur une page séparée du reste du texte.

Il est possible d'associer une légende à une figure ou à une table par la commande `\caption{légende}`. De plus, c'est par le nom `légende` que sera référencé la table ou la figure dans la liste correspondante (voir paragraphe 6.1.).

Enfin, il existe une commande permettant de faire référence dans le texte à la figure ou à la table, c'est la commande `\label{MotRappel}`. Dans le texte on peut alors faire référence à la table par la commande `\ref{MotRappel}` ou même à la page où se trouvera la table par la commande `\pageref{MotRappel}` (voir paragraphe 5.4.).

Code 13

```
\begin{table}
\begin{center}
\begin{tabular}{ll}
18 & 2 \\
5 & 1
\end{tabular}
\caption{Mon premier tableau}
\label{tab1}
\end{center}
\end{table}
Le tableau \ref{tab1} de la page \pageref{tab1} est petit.
```

Résultat 13

18	2
5	1

Tableau 12. Mon premier tableau

Le tableau 12 de la page 19 est petit.

5.4. Références

Comme expliqué dans la section 5.3., pour faire des références, il y a trois commandes.

- 1) La commande `\label{MotRappel}` définit un mot clé `MotRappel` permettant de faire référence à la section, la table ou la figure dans laquelle il est défini.
- 2) La commande `\ref{MotRappel}` renvoie
 - le numéro de la table si le mot clé a été défini dans une table,
 - le numéro de la figure si le mot clé a été défini dans une figure,
 - le numéro de la section si le mot clé a été défini en dehors d'une figure ou d'une table.

- 3) La commande `\pageref{MotRappel}` renvoie le numéro de la page dans laquelle a été défini le mot clé.

Code 14

```
\label{reference}  
On est dans la section \ref{reference} à la page \pageref{reference}
```

Résultat 14

On est dans la section 5.4. à la page 19

Pour que les références apparaissent dans le document final, il faut faire deux compilations successives.

5.5. Annotations

L^AT_EX permet de générer facilement les notes de bas de page par la commande :

```
\footnote{Ma note de bas de page}.
```

La numérotation des notes est automatique.

Code 15

```
le cours de mathématique \footnote{dont le code est MA101} de première année.
```

Résultat 15

le cours de mathématique ² de première année.

Si on désire changer la typographie des appels de notes avec :

- `\arabic` : nombre arabe (1,2,3),
- `\roman` : nombre romain minuscule (i,ii,iii),
- `\Roman` : nombre romain majuscule (I,II,III),
- `\alph` : lettre minuscule (a,b,c),
- `\Alph` : lettre majuscule (A,B,C),
- `\fnsymbol` : symbol (croix, étoile, ...),

la syntaxe se modifie comme suit :

- dans le préambule : `\renewcommand{\thefootnote}{\Alph{footnote}}`
- dans le texte : `\footnote{note de bas de page}`

L'option `\samepage` oblige d'avoir les annotations sur la même page que l'appel.

²dont le code est MA101

6. TABLE DES MATIERES, INDEX ET BIBLIOGRAPHIE

6.1. Table des matières

Les commandes :

- `\tableofcontents`,
- `\listoffigures`,
- `\listoftables`,

insérées à l'endroit du document où on veut placer la table des matières, la liste des figures ou la liste des tableaux suffisent pour les générer. Une double compilation est cependant nécessaire : l'une pour créer la table, l'autre pour l'insérer dans le document.

6.2. Index

Par défaut, `LATEX` ne génère pas d'index pour les documents qui lui sont donnés à compiler. Afin de lui donner cette possibilité, il faut ajouter l'extension `makeidx`.

- `\usepackage{makeidx}` est à mettre dans le préambule,
- `\makeindex` est à mettre dans le préambule,
- `\printindex` est à insérer là où on veut que l'index soit,
- `\index{mot}` est à insérer dans le texte à la suite du mot à mettre dans l'index. Par exemple, l'introduction de `\index{intro}` mettra `intro` dans l'index.

6.3. Bibliographie

Bibtex permet d'imprimer les bibliographies. La première étape est de créer la base de données bibliographiques dans un fichier `biblio.bib`. Dans ce fichier, on encode les différents ouvrages de la bibliographie comme suit :

```
@TypeDeDocument{cle,
  champ1 = "contenu du champ1",
  champ2 = "contenu du champ2",
  champ3 = "contenu du champ3"
}
```

En voici un exemple :

```
@BOOK{Dagn,
  author = {Pierre Dagnelie},
  title = {Statistique théorique et appliquée},
  publisher = {De Boeck & Larcier},
  address = {Paris et Bruxelles},
  year = 1998,
}
```

Comme `TypeDeDocument`, `book` et `article` sont les plus utilisés. Pour les articles, on peut ajouter à `author`, `title`, `journal` et `year` les options `volume`, `number`, `pages`, `month`, `note`.

Dans le corps du texte, les commandes :

- `\bibliographystyle{plain}` précise le style de la bibliographie (`plain` indique que les entrées sont triées alphabétiquement et étiquetées par des numéros entre crochets),
- `\bibliography{biblio}` est placé à l'endroit où la bibliographie doit se trouver.

Lorsqu'on veut citer une référence dans le texte, on utilise le mot de référence mis dans le fichier `biblio.bib` avant `author` avec la commande `\cite{cle}`.

La commande `\nocite{cle}` permet de ne pas mettre l'article se trouvant dans le fichier `biblio.bib` et dont le mot de référence est `cle` dans la bibliographie. La commande `\nocite{*}` met toutes les références du fichier `biblio.bib` dans la bibliographie.

7. DERNIÈRES NOTIONS

Voici trois notions parmi tant d'autres qui peuvent aussi être utiles.

1. Afin d'éviter les orphelins de début ou fin de page, il faut taper dans le préambule

```
\widowpenalty=10000
\clubpenalty=10000
\raggedbottom
```

ou utiliser la commande `\enlargethispage` qui impose à $\text{\LaTeX}$ de compresser ou d'étendre le contenu d'une page afin d'éviter que la page suivante contienne trop peu de texte.

2. Afin de ne pas avoir un fichier source trop grand, on peut découper le fichier source en plusieurs fichiers. La commande `\include{intro}` intègre le fichier `intro.tex` sur une nouvelle page, à l'endroit où se trouve cette commande. Pour intégrer un fichier directement dans la suite du texte, et non sur une nouvelle page, il faut utiliser la commande `\input{intro}`. La commande `\includeonly{intro}` ne compile que le fichier `intro.tex`. Cette commande est à placer dans le préambule.

3. La césure des mots faite par $\text{\LaTeX}$ peut être différente de ce qui est souhaité. Dès lors, la commande `gé\~né\~ra\~li\~sa\~tion` placée dans le texte indique à $\text{\LaTeX}$ qu'il peut couper le mot généralisation à l'endroit où se trouve `\~`.

La commande `\hyphenation{er-go-no-mic}` placée dans le préambule concerne les mots sans accent et indique à $\text{\LaTeX}$ qu'il peut couper le mot ergonomic là où se trouve les `-` chaque fois qu'il rencontre ce mot dans le texte.

Enfin, pour obtenir de plus amples informations, il est conseillé de consulter les références se trouvant dans la bibliographie ou sur le site <http://www.grappa.univ-lille3.fr/FAQ-LaTeX/>.

BIBLIOGRAPHIE

- BAYART B. [1995]. *Joli manuel pour LaTeX_{2 ϵ}* . ESIEE, 143p.
- KNUTH D. [1986]. *The TeXbook*. Addison-Wesley, 496p.
- LAMPORT L. [1994]. *Latex : a document preparation system. User's guide and reference*. Addison-Wesley.
- MITTELBACH F., GOOSSENS M. [2005]. *Latex Companion*. Pearson Education.
- OETIKER T., PARTL H., HYNÄ I., SCHLEGL E. [2007]. *The Not So Short Introduction to LaTeX_{2 ϵ}* .
- ROLLAND C. [1999]. *L^AT_EXpar la pratique*. O'Reilly, Paris, 559p.