

Introduction à L^AT_EX

Damien MATEO Eté 2001

Table des matières

1	Qu'est ce que L^AT_EX et pourquoi l'utiliser	4
2	Comment faire du L^AT_EX sous Windows	4
3	Principe et Vocabulaire	4
3.1	Principe	4
3.2	Vocabulaire	5
4	Les Packages	5
5	Structure d'un document L^AT_EX	7
5.1	Architecture du document principal : <i>fichier.tex</i>	7
5.2	Structure hiérarchique	7
5.3	Insertion de fichiers	7
6	Les Caractères	8
6.1	Caractères de base	8
6.2	Caractères réservés	8
6.3	Caractères de commandes	8
6.4	Caractères spéciaux	8
7	Les Commandes	8
7.1	Définition de commandes	9
7.2	Commandes usuelles	9
7.3	Fonts et styles	9
7.4	Tailles	10
8	Les Environnements	10
8.1	Les environnements prédéfinis	10
8.1.1	Les alignements	10
8.1.2	Les listes	10
8.1.3	Les tableaux	11
8.1.4	L'environnement <i>theorem</i>	12
8.1.5	L'environnement <i>minipage</i>	13
8.1.6	L'environnement <i>em</i>	13
8.1.7	Autres environnements	14
8.2	Les environnements définis par l'utilisateur	14
9	Mise en page	14
9.1	Espacements et sauts	14
9.2	En-têtes et pieds de pages	15
9.3	Les références croisées	15
10	Comment faire un index	16

11 Figures, Tables et Graphisme	16
11.1 Tables et Figures	16
11.2 Inclusion d'images au format <i>eps</i>	17
11.2.1 Comment obtenir un fichier au format <i>eps</i>	17
11.2.2 Comment procéder sous L ^A T _E X	17
12 Mathématiques	19
12.1 Les environnements mathématiques	19
12.2 Les polices mathématiques	20
12.3 Les règles	20
12.4 Eléments d'une formule	20
13 Utilisation des packages	23
13.1 Comment installer les packages optionnels	23
13.2 L'extension <i>color</i>	23
13.3 L'extension <i>fancybox</i>	23
13.3.1 Les boîtes de base	24
13.3.2 Les boîtes avec environnements	24
13.4 L'extension <i>fancyhdr</i>	25
13.5 L'extension <i>fancyvrb</i>	27
13.6 L'extension <i>listings</i>	28
13.7 L'extension <i>pb-diagram</i>	30
14 Tableaux de commandes	32

1 Qu'est ce que \LaTeX et pourquoi l'utiliser

\LaTeX n'est pas un traitement de texte "Ce que vous voyez est ce que vous tapez" comme Word. Etablir un document \LaTeX c'est écrire un script contenant à la fois le texte et des commandes de mise en page et de mise en forme de caractères. On peut en outre spécifier la taille des pages et insérer aisément des formules mathématiques.

L'avantage majeur de ce traitement de texte est qu'il est un véritable langage de programmation. L'utilisateur peut lui même définir ses propres commandes pour automatiser la mise en forme de son document. De plus, le compilateur \LaTeX a été porté sur une grande variété de plate-formes (Windows, Mac, Solaris, Linux...). Il est possible de trouver des distributions gratuites sur chacun de ces environnements (Windows : MikTeX, FpTeX... Linux : TeTeX,...). Enfin, une source \LaTeX requiert infiniment moins d'espace mémoire qu'un fichier généré par Word.

\LaTeX se distingue des autres éditeurs de texte quant à la mise en page soignée des documents qu'il génère et la souplesse de la mise en forme. L'utilisateur a donc la possibilité de concentrer toute son attention sur le contenu de son texte.

Le document que voici est orienté vers une utilisation sous un environnement *Windows*, avec la batterie de compilateurs *MikTeX*.

2 Comment faire du \LaTeX sous *Windows*

Pour pouvoir faire du \LaTeX chez vous, il vous suffit de vous procurer la dernière version de *MikTeX*, un éditeur (*Winedit*, *Winshell*, *Emacs...*), ainsi qu'un visualiseur de fichier *post-script* (*Ghostview*). On peut trouver tous ces logiciels aux adresses suivantes

MikTeX (freeware) Site officiel

ou bien à l'adresse CTAN.

La dernière version recensée est la 2.1.

Winedit (shareware) Site officiel ou bien à l'adresse

CTAN

Winshell (freeware) Site officiel ou bien à l'adresse

Université de Poitiers

Ghostview (freeware) *Ghostscript* et *Ghostview* à l'adresse Site officiel

Remarque 1 Sous Linux, tous les packages rpm pour faire du \LaTeX sont disponibles sur les CD d'installation si vous ne les avez pas sélectionnés lors de l'installation de Linux.

3 Principe et Vocabulaire

3.1 Principe

La création d'un document en \LaTeX est composée de plusieurs étapes. Il faut tout d'abord écrire un *fichier d'entrée* (extension *.tex*) qui contiendra le texte, les commandes de mise en page et de mise en forme. Ensuite, ce fichier est compilé en un *fichier dvi* destiné à être prévisualisé. Il crée pour cela plusieurs fichiers :

- un *fichier log* qui contient une liste d'erreurs de compilation ainsi que des commentaires
- un *fichier aux* contenant les informations nécessaires à la gestion des titres, des définitions, des étiquettes...

De même, on remarque qu'il sera créé

- un *fichier toc* si l'utilisateur demande la création d'une table des matières
- un *fichier idx* si l'utilisateur demande la création d'un index.

Le *fichier dvi* donne un aperçu du document tel qu'il sera imprimé par la suite.

Ce fichier nécessite de nombreuses polices de caractères. Si on veut le visualiser sur un autre ordinateur, il faut impérativement disposer de toutes ces polices, ce qui n'est pas toujours le cas. De plus il faut avoir tous les fichiers que l'on peut inclure dans le document (des images par exemple). Pour remédier à ce problème, il est possible de convertir le *fichier dvi* en un *fichier ps (PostScript)*, format qui est reconnu sur tout ordinateur, du moment qu'il possède un logiciel de visualisation adéquat. L'inconvénient du *PostScript* est la taille plutôt volumineuse du fichier généré (de l'ordre de la centaine de Ko).

Remarque 2 *L'éditeur de texte, sous lequel l'utilisateur va créer son document L^AT_EX, permet souvent d'exécuter automatiquement les tâches de compilation, prévisualisation dvi, et conversion PostScript au moyen d'icônes cliquables.*

3.2 Vocabulaire

Préambule : C'est la partie située entre la première instruction d'un document et le début de l'environnement principal.

Commande : Ce sont les instructions de base de L^AT_EX qui permettent de changer des paramètres de mise en forme du texte ou l'insertion d'une partie de texte mémorisée. Les commandes sont identifiées par des noms qui commencent tous par un `\`.

Environnement : Ils permettent de mémoriser une mise en forme de texte et de l'appliquer ainsi à plusieurs endroits du document. Il existe des environnements prédéfinis pour créer des listes, des tableaux, des matrices, théorèmes... mais il est également possible de définir ses propres environnements. La modification de la définition d'un environnement entraîne la modification de la mise en forme du texte partout où l'on a utilisé cet environnement.

Théorème : Afin de regrouper les structures similaires telles que les théorèmes, les définitions, les lemmes... L^AT_EX propose un environnement générique qui permet de dériver ces structures avec ajout de numérotations.

Mode Texte : On dit que L^AT_EX est en mode texte s'il compile une portion de texte comme un paragraphe, un tableau, une énumération... S'il n'est pas en mode texte, il est en mode mathématique. Certaines instructions n'y sont donc pas applicables.

Mode Mathématique L^AT_EX se trouve en mode mathématique s'il compile une formule mathématique. Il utilise une mise en forme totalement différente de celle du mode texte. Par exemple, il ne respecte pas les espaces.

Les longueurs : On compose un document L^AT_EX avec des commandes qui demandent souvent une taille ou une longueur (espacements, tailles de fenêtres...). Il est donc important de connaître les unités de base :

pt	le <i>point</i>	1pt=0.351mm
mm	le <i>millimètre</i>	1mm=2.845pt
cm	le <i>centimètre</i>	1cm=10mm
in	le <i>pouce</i>	1in=2.54cm

4 Les Packages

Les extensions (*packages* en anglais) servent à modifier la mise en page ou à définir de nouvelles commandes. Ce sont elles qui permettent d'ajouter de nouvelles fonctions à L^AT_EX. Elles sont chargées grâce à

la commande `\usepackage`, utilisée dans le préambule du document. Certains de ces packages sont livrés d'origine avec L^AT_EX, d'autres doivent être installés. Les déclarations se font dans le préambule.

Parmi les packages standards on trouve :

- `amsmath`, `amsfonts`, `amssymb` qui sont des extensions de l'*American Mathematical Society* relatives aux polices et insertions de formules mathématiques. Il faut insérer dans le préambule :
`\usepackage{amsmath}`, `\usepackage{amsfonts}` et `\usepackage{amssymb}`
- `array` qui est une extension des environnements prédéfinis `array` et `tabular` : `\usepackage{array}`
- `babel` qui renomme les noms de chapitre, dates et autres textes insérés par L^AT_EX dans la langue choisie. `babel` accorde aussi la typographie aux règles en vigueur selon les pays. Il faut insérer dans le préambule : `\usepackage[french]{babel}`
- `tlenc` et `inputenc` qui permettent de disposer de lettres accentuées sans avoir à les appeler par des commandes. Il faut insérer dans le préambule :
`\usepackage{tlenc}`, et `\usepackage[latin1]{inputenc}`
- `color` pour disposer des couleurs : `\usepackage{color}`
- `graphics` et `graphicx` pour la manipulation de boîtes (encadrés) et l'insertion d'images `\usepackage{graphics}` et `\usepackagegraphicx`
- `makeidx` pour la création d'un index : `\usepackage{makeidx}`.

Parmi les packages optionnels on trouve :

- `fancybox` qui est une extension du lot de boîtes prédéfinies : `\usepackage{fancybox}`
- `fancyvrb` qui est une extension de l'environnement standard `Verbatim` : `\usepackage{fancyvrb}`
- `fancyhdr` qui est une extension des en-têtes en pieds-de-page. Ce package permet de les composer soi-même. Il faut insérer dans le préambule : `\usepackage{fancyhdr}`
- `listings` qui permet d'insérer des codes sources. Il permet par exemple de mettre en évidence les instructions propres au langage (Matlab, C++...)
- `pb-diagram` pour faire de beaux diagrams avec des flèches indexées.

Un nombre très important de packages peuvent être téléchargés sur le site du *Loria* à l'adresse suivante :

.

Pour TOUT renseignement concernant L^AT_EX, une encyclopédie est également disponible sur le site du *Loria* à l'adresse :

.

En fin de document, j'expliquerai comment installer un package optionnel. De plus, je présenterai succinctement l'utilisation des extensions les plus utiles (les optionnelles principalement).

5 Structure d'un document L^AT_EX

5.1 Architecture du document principal : *fichier.tex*

```
\documentclass[Options]{Style}

Préambule {
  \usepackage{t1enc}
  \usepackage[french]{babel}
  ...autres packages...
  \title{Introduction à \LaTeX}
  \author{Damien}
  \date{17 juillet 2001}
  ...commandes et paramètres propres au document...

\begin{document}
  ...Corps du document...
\end{document}
```

Options : *10pt, 11pt, 12pt, a4paper, french, draft, twocolumn, twoside.*

Style : *article, report, letter, book, slides.*

5.2 Structure hiérarchique

	<i>article</i>	<i>report</i>	<i>book</i>
<code>\part{titre}</code>	O	X	X
<code>\chapter{}</code>	O	X	X
<code>\section{}</code>	X	X	X
<code>\subsection{}</code>	X	X	X
<code>\subsubsection{}</code>	X	X	X
<code>\paragraph{}</code>	X	X	X
<code>\subparagraph{}</code>	X	X	X

Remarque 3 Dans la table des matières, l'apparition des niveaux hiérarchiques est de trois par défaut pour un article (jusqu'à `\subsubsection`) et de quatre pour les rapports et les livres (jusqu'à `\subsection`). Il est possible d'aller plus en profondeur dans la numérotation en utilisant dans le préambule la commande : `\setcounter{secnumdepth}{i}` où *i* est le niveau de profondeur.

Remarque 4 Les formes étoilées de ces commandes ne sont pas numérotées et ne créent pas d'entrées dans la table des matières (`\section*{}` par exemple).

Remarque 5 Conceptuellement, la différence entre la classe `book` et les autres est qu'elle est conçue pour être une succession de feuilles recto-verso, alors que les autres sont des successions de feuilles recto.

Textuellement, la différence se ressent en niveau des options de mise en page (en-têtes et pieds de page) Je vous invite à la section ?pour plus d'explications.

5.3 Insertion de fichiers

Pour plus de lisibilité au sein du script, on peut découper son document L^AT_EX en plusieurs sources à insérer dans le document final avec les commandes

- `\input{fichier}` cette commande est remplacée par le contenu de *fichier.tex*. Le fichier *fichier* peut lui-même inclure une commande `\input`
- `\include{fichier}` cette commande insère le fichier *fichier.tex*. Le fichier ne sera pas recompilé s'il n'a pas été modifié depuis la dernière compilation. `\include` ne peut apparaître dans le préambule et le texte doit former un ensemble de pages indépendantes
- `\includeonly{fichier1, fichier2...}` placée dans le préambule, cette commande restreint la liste des fichiers insérés à l'aide de la commande `\include`. Les fichiers absents de la liste ne seront pas remis à jour même s'ils ont été modifiés depuis la dernière compilation.

Remarque 6 : *La grande utilité réside en une visualisation plus aérée (pour ce document, j'ai réalisé chaque section dans un fichier portant le nom de la section, puis je les ai toutes réunies dans un document ne contenant que le préambule, les déclarations de commandes et environnements, et les commandes d'insertions `\include` dans l'environnement document). Le gain certain se fait ressentir au niveau du temps de compilation, quand le compilateur ne retouche que les fichiers modifiés.*

6 Les Caractères

6.1 Caractères de base

Sont autorisés dans un document \LaTeX les minuscules, les majuscules, les chiffres et les caractères suivants :

. ; , ? : ! ' ' " () [] @ * / - + > <

6.2 Caractères réservés

Caractères réservés par \LaTeX pour définir ses propres commandes :

\$ & \% _ ^ { }

6.3 Caractères de commandes

\$	Mode mathématique	~	Espace insécable	^ et _	exposant et indice
%	Commentaire	&	Alignement de tableau	{...}	groupe

6.4 Caractères spéciaux

†	\dag	¶	\P	#	\#	\	\backslash	{	\{	§	\S	£	\pounds
‡	\ddag	&	\&	\$	\\$	©	\copyright	}	\}	_	_	%	\%

7 Les Commandes

La commande constitue l'un des principaux composants de \LaTeX . Chaque commande est identifiée par un nom unique qui doit commencer par `\` et se poursuivre par une séquence de lettres. \LaTeX distingue les majuscules et les minuscules. Des tableaux de commandes usuelles sont disponibles en fin de document.

Exemples La commande `\LaTeX` renvoie à l'écran \LaTeX . La commande `\heartsuit` renvoie ♥.

7.1 Définition de commandes

L'utilisateur a la possibilité de définir ses propres commandes lui permettant de taper plus rapidement les séquences de commandes les plus récurrentes :

- `\newcommand{ \nom}[nb arguments]{définition}` : définit une nouvelle commande
- `\renewcommand` : redéfinit ou définit une commande
- `\providecommand` : définit une commande sauf si elle existe déjà.

Exemples

- `\newcommand{ \anti}{anticonstitutionnellement}` : définit une nouvelle commande `\anti` qui à tout appel renverra à l'écran le mot *anticonstitutionnellement*
- `\newcommand{\seq}[1]{\${#1}_1, \ldots, {#1}_n}` : définit une nouvelle commande `\seq` qui à tout appel `\seq{x}` renverra à l'écran x_1, \dots, x_n . `\seq{y}` renverra y_1, \dots, y_n .

7.2 Commandes usuelles

<code>\maketitle</code>	produit un titre avec l'auteur et la date, tous trois spécifiés dans le préambule
<code>\tableofcontents</code>	produit automatiquement une table des matières suivant la hiérarchisation du document
<code>\TeX \LaTeX \LaTeXe</code>	$\TeX \LaTeX \LaTeX 2_\epsilon$
<code>\Verb"..."</code>	mode verbatim (affiche à l'écran rigoureusement tout ce que vous avez mis entre guillemets en police type machine à écrire)
<code>\begin{env}...\end{env}</code>	bloc inclus dans un environnement <i>env</i>
<code>\footnote{note}</code>	note de bas de page
<code>_</code>	provoque un espace
<code>\\</code> ou <code>\newline</code>	provoque un saut de ligne
<code>\newpage</code>	provoque un saut de page
<code>\marginpar{note}</code>	produit une <i>note</i> dans la marge de droite
<code>\printindex</code>	produit un index à l'endroit où se trouve la commande Voir la partie "Comment faire un index".

7.3 Fonts et styles

Texte isolé	Changement de police	Exemple	Commentaire
<code>\textrm{texte}</code>	<code>\rmfamily</code>	ABCabc123	fonte de type roman par défaut
<code>\textsf{texte}</code>	<code>\sffamily</code>	ABCabc123	fonte de type sans serif
<code>\texttt{texte}</code>	<code>\ttfamily</code>	ABCabc123	fonte de type type writer
<code>\emph{texte}</code>		<i>ABCabc123</i>	mise en <i>évidence</i> d'une portion de texte
<code>\textbf{texte}</code>	<code>\bfseries</code>	ABCabc123	corps gras
<code>\textmd{texte}</code>	<code>\mdseries</code>	ABCabc123	corps moyen par défaut
<code>\textit{texte}</code>	<code>\itshape</code>	<i>ABCabc123</i>	forme <i>italique</i>
<code>\textsl{texte}</code>	<code>\slshape</code>	<i>ABCabc123</i>	forme <i>penchée</i>
<code>\textsc{texte}</code>	<code>\scshape</code>	ABCABC123	forme PETITES MAJUSCULES
<code>\textup{texte}</code>	<code>\upshape</code>	ABCabc123	forme droite par défaut

7.4 Tailles

<code>\tiny</code>	Texte	<code>\large</code>	Texte
<code>\scriptsize</code>	Texte	<code>\Large</code>	Texte
<code>\footnotesize</code>	Texte	<code>\LARGE</code>	Texte
<code>\small</code>	Texte	<code>\huge</code>	Texte
<code>\normalsize</code>	Texte	<code>\Huge</code>	Texte

Remarque 7 On peut souligner du texte en utilisant la commande `\underline{texte}`.

8 Les Environnements

Les environnements sont les fondements de la mise en page \LaTeX . Ils permettent de spécifier une présentation particulière du texte. Pour cela, il faut placer une instruction en indiquant le début et une autre pour la fin. Un environnement est identifié par un nom. \LaTeX possède de nombreux environnements prédéfinis qui permettent de créer des listes numérotées ou non, des citations, des textes littéraux, des tableaux... Pour un environnement *env* donné, on procède comme suit :

<code>\begin{env}</code> <i>texte</i> <code>\end{env}</code>	\LaTeX présentera le <i>texte</i> avec la mise en page spécifiée dans l'environnement.
--	---

Remarque 8 Le premier environnement abordé est l'environnement `document`. Il est indispensable à la rédaction d'un document \LaTeX .

8.1 Les environnements prédéfinis

8.1.1 Les alignements

Il existe trois environnements pour aligner les textes

<code>\begin{flushleft}</code> <i>texte aligné</i> à gauche <code>\end{flushright}</code>	<code>\begin{center}</code> <i>texte</i> centré <code>\end{center}</code>	<code>\begin{flushright}</code> <i>texte aligné</i> à droite <code>\end{flushright}</code>
--	--	---

Remarque 9 Les commandes `\raggedright`, `\centering`, et `\raggedleft` changent la pagination du document en entier en alignant respectivement tout à gauche, centré et à droite. Par défaut, \LaTeX est en mode justifié (aligné à droite et à gauche).

8.1.2 Les listes

\LaTeX propose trois environnements basés sur les listes, de construction identique.

Listes de puces Les puces par défaut sont les tirets

```
\begin{itemize}
\item un           - un
\item deux        - deux
\end{itemize}
```

Mais elles sont paramétrables

```
\begin{itemize}
\item[$\dagger$] un      † un
\item[$\dagger$] deux    † deux
\end{itemize}
```

Listes par énumération avec numérotation

```
\begin{enumerate}
\item un           1. un
\item deux        2. deux
\end{enumerate}
```

Listes descriptives avec retrait de texte

```
\begin{description}
\item[un :] première description      un : première description
\item[deux :] deuxième description   deux : deuxième description
\end{description}
```

Remarque 10 *les environnements de listes sont imbriquables. \LaTeX change alors automatiquement les puces ou numéros pour être cohérent.*

8.1.3 Les tableaux

Pour mettre en forme des tableaux avec \LaTeX , il existe deux environnements totalement similaires : *tabular* et *array*. La différence est due à l'utilisation que l'on fait :

tabular : utilisé en mode texte pour faire des tableaux dont les cellules sont en mode texte

array : utilisé en mode mathématique pour créer des matrices

Je vais présenter la manipulation de *tabular*. Nous reviendrons sur la présentation d'*array* lorsque je présenterai le mode mathématique. La syntaxe d'un tableau est la suivante

```
\begin{tabular}[position]{format colonnes}
Case(1,1) & Case(1,2) & ... \\
Case(2,1) & Case(2,2) & ... \\
...
\end{tabular}
```

où

position(optionnel)	signification
t	tableau aligné avec la ligne du haut
b	tableau aligné avec la ligne du bas

format colonnes	signification
l	colonne alignée à gauche
c	colonne centrée
r	colonne alignée à droite
	provoque une ligne verticale entre deux colonnes
	provoque une double ligne verticale entre deux colonnes

Nous avons de plus les commandes suivantes

- `\multicolumn{nombre}{format colonne}{texte}` fait de *texte* l'élément d'une cellule qui prend la place de *nombre* cellules dans la ligne où elle se trouve. *format colonne* contient les mêmes instructions que *format colonnes*, mais pour une seule colonne
- `\vline` provoque une ligne verticale de la hauteur de la cellule dans une colonne
- `\hline` provoque une ligne horizontale sur la largeur du tableau
- `\cline{i-j}` provoque une ligne horizontale de la colonne *i* à *j*.

Remarque 11 *Les tableaux sont imbricables à volonté.*

Voici un exemple complet

```

\begin{tabular}{|l|c|c|c|}
\hline
& \multicolumn{3}{|c|}{vente} \\
\cline{2-4}
\multicolumn{1}{|c|}{\raisebox{6pt}[0pt][0pt]{Fruit}} & & & \\
& 1995 & 1996 & 1997 \\
\hline
Poire & 6.5 & 7.8 & 7.2 \\
Pomme & 15 & 16 & 15.5 \\
\hline
\end{tabular}

```

Fruit	vente		
	1995	1996	1997
Poire	6.5	7.8	7.2
Pomme	15	16	15.5

8.1.4 L'environnement *theorem*

Les textes mathématiques contiennent souvent des théorèmes et des structures similaires telles que axiomes, définitions, lemmes, propositions... Avoir un environnement particulier pour chaque structure ne serait pas pratique. L^AT_EX propose la commande

```
\newtheorem{nom d'appel}[nom du compteur associé]{nom affiché}
```

Exemples

Dans le préambule	Appel	Aperçu
<code>\newtheorem{theo}{Théorème}</code>	<code>\begin{theo}</code> <i>texte</i> <code>\end{theo}</code>	Théorème 1 <i>texte</i>
<code>\newtheorem{prop}[theo]{Proposition}</code>	<code>\begin{prop}[Dam]</code> <i>texte</i> <code>\end{prop}</code>	Proposition 2 (Dam) <i>texte</i>
<code>\newtheorem{lem}[theo]{Lemme}</code>	<code>\begin{lem}</code> <i>texte</i> <code>\end{lem}</code>	Lemme 3 <i>texte</i>

Remarque 12 *L'environnement défini est theo. Pour les déclarations des autres environnements (prop et lem), on spécifie la synchronisation de leur compteur avec celui de theo en indiquant [theo]. On peut faire la même opération en liant le compteur au chapitrage ([chapter] ou [section]). On peut de même donner une indication sur le théorème en la spécifiant entre crochets ([Dam]) sur la ligne du `\begin`.*

Remarque 13 *On peut travailler sur les compteurs avec les deux commandes suivantes*

- `\stepcounter{compteur}` : va incrémenter le compteur `compteur` de un
- `\setcounter{compteur}{i}` : va initialiser le compteur `compteur` à la valeur `i`.

8.1.5 L'environnement *minipage*

Il arrive que l'on veuille créer des portions de texte assez complexes (sauts de lignes, insertion de listes...) dans des tableaux ou des boîtes. Il faut alors utiliser un environnement spécifique qui permet une utilisation sans conflits : c'est l'environnement *minipage*.

```
\begin{minipage}[position]{largeur}
instructions et commandes
\end{minipage}
```

position :

- centré par défaut (sans crochets)
- `t` : ligne supérieure alignée avec la ligne courante du texte
- `b` : ligne inférieure alignée avec la ligne courante du texte

largeur : par exemple : 3cm, 5cm...

8.1.6 L'environnement *em*

L'environnement *em* met son contenu en style `\emph`. Cela permet de mettre en relief plusieurs paragraphes.

```
\begin{em}
texte en relief
\end{em}
```

affichera *texte en relief*

8.1.7 Autres environnements

abstract		résumé
letter{ <i>destinataire</i> }		en classe document letter, lettre indépendante
picture		environnement de dessin
quotation		citation avec indentation des paragraphes
quote		citation sans indentation des paragraphes
titlepage		page de titre
verbatim		imprime en type machine à écrire un bloc non interprété
verbatim*		idem verbatim sauf que les espaces sont marqués ◻
verse		poésie

8.2 Les environnements définis par l'utilisateur

L'utilisateur a la possibilité de définir ses propres environnements lui permettant de taper plus rapidement les compositions de texte les plus récurrentes :

- `\newenvironment{env}[nb arguments]{texte debut}{texte fin}` : définit un nouvel environnement *env*
- `\renewenvironment` : redéfinit ou définit un environnement

Exemple `\newenvironment{dem}{Démonstration}{ \diamond \diamondsuit}` : définit un nouvel environnement *dem* qui à tout appel

<code>\begin{dem}</code>		Démonstration
<i>texte</i>	affichera	<i>texte</i>
<code>\end{dem}</code>		◊

9 Mise en page

9.1 Espacements et sauts

~,	espace	<code>\-</code>	emplacement de césure	<code>\\[h]</code>	retour à la ligne, <i>h</i> est l'interligne
	petit espace	<code>\/</code>	espace après italique	<code>*[h]</code>	retour à la ligne sans saut de page
	espace insécable			<code>\newline</code>	saut de ligne

Commande	Utilité
<code>\hspace{l}</code>	espace horizontal, ignoré en cas de saut de ligne
<code>\vspace{h}</code>	espace vertical, ignoré en cas de saut de ligne
<code>\hspace*{l}</code>	espace horizontal
<code>\vspace*{h}</code>	espace vertical
<code>\hfill</code>	espace élastique horizontal
<code>\vfill</code>	espace élastique vertical
<code>\hrule fill</code>	ligne élastique horizontale
<code>\dotfill</code>	points élastiques horizontaux
<code>\smallskip</code>	petit espace vertical
<code>\medskip</code>	moyen espace vertical
<code>\bigskip</code>	grand espace vertical

Commande	Utilité	Exemples pour ce document
<code>\baselineskip</code>	espace entre les lignes	<code>\baselineskip 12pt</code>
<code>\textwidth</code>	espace sur lequel le texte va être ajusté (depuis la marge de gauche)	<code>\textwidth 16.2cm</code>
<code>\oddsidemargin</code>	modification de la largeur de la marge de gauche par rapport à la valeur par défaut pour les pages impaires	<code>\oddsidemargin -0.3cm</code>
<code>\evensidemargin</code>	modification de la largeur de la marge de gauche par rapport à la valeur par défaut pour les pages paires	<code>\evensidemargin -0.3cm</code>
<code>\parskip</code>	espace entre les paragraphes	<code>\parskip 0pt</code>

9.2 En-têtes et pieds de pages

Il existe dans L^AT_EX des commandes permettant de gérer soi-même la mise en tête ou pied de page. On distingue deux commandes :

`\pagestyle{style}` : dans le préambule, définit le style de page du document entier

`\thispagestyle{style}` : définit le style de la page courante.

L'argument *style* peut prendre principalement quatre valeurs différentes :

- **plain** est le style par défaut : pas d'en-tête, seulement la numérotation centrée pour pied de page
- **empty** pour n'avoir ni en-tête ni pied de page
- **headings** pour avoir en en-tête le chapitrage et la pagination, rien en bas de page
- **fancy** pour définir soi-même ses en-têtes et pieds de page (voir la section sur les compléments d'utilisation du package *fancyhdr*).

Remarque 14 On peut changer le style de numérotation des pages en utilisant la commande `\pagenumbering{style}`. A noter que la définition du style peut être définie pour n'importe quel compteur. Nous avons laissé le compteur de *theo* à 3 (section ?? page ??). L'argument *style* peut prendre les quatre valeurs suivantes :

style	exemple avec le compteur de <i>theo</i>
<i>arabic</i>	3
<i>roman</i>	iii
<i>Roman</i>	III
<i>alph</i>	c
<i>Alph</i>	C

9.3 Les références croisées

L^AT_EX possède des commandes pour renvoyer le numéro de n'importe quel titre, équation ou page, pourvu qu'on l'ai étiqueté.

Procédons par l'exemple.

Imaginons que nous ayons le 3ème chapitre : **III Analyse Numérique**.

Imaginons que nous voulions y faire une référence : ...comme indiqué au indiqué au chapitre 3 page 17.

Mais l'avantage de L^AT_EX est justement de pouvoir insérer des chapitre et sections où bon nous semble et les numéros de chapitres ne sont pas fixes. Pour que les liens soient faits lors des références, il faut étiqueter le chapitre avec la commande `\label` :

```
\chapter{Analyse Numérique} \label{AL}
```

et on renvoie les informations avec les commandes `\ref` et `\pageref` :

...comme indiqué au chapitre `\ref{AL}` page `\pageref{AL}`.

C'est à dire que dans le script, tout appel de `\ref{AL}` renverra **III** et tout appel de `\pageref{AL}` renverra 17.

Remarque 15 *Plusieurs compilations peuvent être nécessaires pour construire les références.*

10 Comment faire un index

On peut faire des index aisément sous \LaTeX . Il faut insérer les deux lignes de commande suivantes dans le préambule

```
- \usepackage{makeidx}
- \makeindex
```

Dès que l'on veut mettre un mot dans l'index, il faut insérer `\index{mot}` juste après le mot dans la source. Par exemple, si on parle de la fonction *cosinus*, et qu'on veut la mettre dans l'index, on insère la ligne de commande `\index{cosinus}`. Même mieux, si on veut la mettre dans la catégorie *fonctions*, on insère la commande `\index{fonctions!cosinus}`. Ensuite, on met `\printindex` à l'endroit même où on veut voir apparaître l'index dans le document.

Ensuite, tout se fait en trois étapes.

1. Il faut compiler *fichier.tex* avec \LaTeX une première fois. Un fichier *fichier.idx* est créé. Il contient toutes les informations nécessaires à la construction d'un index.
2. Il faut ouvrir une boîte *DOS* (sous *Windows*), se placer dans le répertoire courant et entrer la ligne de commande

```
makeindex fichier.idx
```

Un fichier *fichier.ind* est alors créé. Il sera automatiquement inclus dans la prochaine compilation \LaTeX de *fichier.tex*. Nous abordons alors la dernière étape.

3. Il reste à compiler deux fois *fichier.tex* pour que les modifications avec l'index soient prises en compte.

11 Figures, Tables et Graphisme

11.1 Tables et Figures

Les tables et figures sont des environnements flottants. Ces environnements sont numérotés et placés de manière différente du reste du texte (\LaTeX cherche à optimiser la mise en page pour éviter des blancs dans le texte). La syntaxe est la suivante

<code>\begin{figure}[position] ... \end{figure}</code>	figure (généralement un graphe ou une image)
<code>\begin{table}[position] ... \end{table}</code>	table (généralement un tableau)
<code>\caption{légende}</code>	produit une <i>légende</i> (dans une figure ou une table)

où *position* est une suite de lettres décrivant le placement souhaité pour l'objet flottant

position(optionnel)	signification
<i>h</i>	<i>here</i> : la figure est placée là où elle apparaît dans le script
<i>t</i>	<i>top</i> : la figure est placée en haut d'une page de texte
<i>b</i>	<i>bottom</i> : la figure est placée en bas d'une page de texte
<i>p</i>	<i>page of floats</i> : la figure est placée sur une page ne contenant que des flottants

Remarque 16 De manière analogue à la création de la table des matières avec la commande `\tableofcontents`, on peut générer la liste des tables et des figures avec les commandes `\listoftables` et `\listoffigures`.

11.2 Inclusion d'images au format *eps*

L'inclusion, ou insertion, d'images est très pratique pour présenter des graphes.

11.2.1 Comment obtenir un fichier au format *eps*

Sous *Matlab*, on peut convertir un graphe au format *eps* de deux manières différentes

- Sur la fenêtre graphique, cliquez sur *file->export->type EPS level 2 color (*eps)*
- Dans l'espace *Matlab*, entrez `>>print -depsc2 fichier.eps`

Remarque 17 Il existe de petits exécutables pour convertir des fichiers au format *jpg* en fichier *eps*. Ils sont disponibles sur le site du Loria (adresse en début de document).

11.2.2 Comment procéder sous \LaTeX

Dans un premier temps, il faut spécifier le *path* dans le préambule :

```
\graphicspath{chemin 1}{chemin 2}...
```

Exemples

- `\graphicspath{./}` spécifie qu'il faut aller chercher les fichiers au même niveau de l'arborescence que le document \LaTeX lui-même
- `\graphicspath{./images}` spécifie qu'il faut aller chercher les fichiers dans le répertoire *images*, situé au même niveau de l'arborescence que le document \LaTeX lui-même.

Alors l'insertion du graphe dans le script se fait par la commande

```
\includegraphics[options]{fichier.eps}
```

où [*options*] peut être par exemple [*angle=90,width=10cm*]. L'image sera alors tournée de 90 degrés dans le sens inverse des aiguilles d'une horloge, puis ajustée pour que sa largeur finale soit de 10 cm. Cela marche également avec l'argument *height*.

Remarque 18 Il est très utile de définir la commande suivante dans le préambule

```
\newcommand{\imageps}[2]{
\begin{center} \hspace{-0.5cm} \includegraphics[scale={#2}]{#1}
\end{center}}
```

Elle permet d'afficher un graphe centré dont la taille est proportionnelle au second argument de la commande.

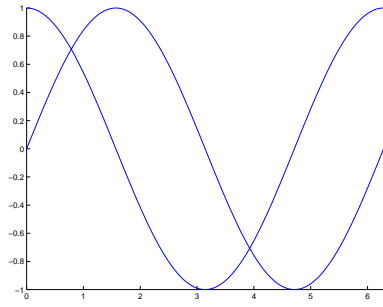


FIG. 1 – `\imageps{graphe.eps}{.3}`

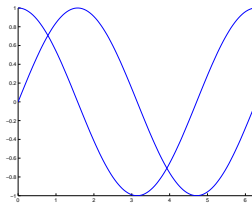


FIG. 2 – `\imageps{graphe.eps}{.2}`

Exemples

12 Mathématiques

12.1 Les environnements mathématiques

L^AT_EX permet de générer des formules mathématiques du meilleur goût ! Il faut distinguer pour cela la forme désirée

Expression mathématique intégrée à un paragraphe	\dots \dots \dots
Expression mathématique isolée	\dots
Equation numérotée en mode <code>\displaymath</code> Equation non numérotée en mode <code>\displaymath</code>	\dots \dots
Système d'équation numérotées Système d'équation non numérotées	\dots \dots

`$$ a(x) = b $$`

Exemples

Source	Aperçu
<code>100~m\$^{3}\$ d'eau (H\$_{2}\$O)</code>	100 m ³ d'eau (H ₂ O)
<code>\$\$\overbrace{a+b+c}^{d}\$\$</code>	$\overbrace{a+b+c}^d$
<code>\$\int_{\Omega_0} \zeta(\omega) d\omega \geq \bar{r}\$</code>	$\int_{\Omega_0} \zeta(\omega) d\omega \geq \bar{r}$
<code>\$\$\int_{\Omega_0} \zeta(\omega) d\omega \geq \bar{r}\$\$</code>	$\int_{\Omega_0} \zeta(\omega) d\omega \geq \bar{r}$
<code>\begin{equation} \epsilon > 0 \end{equation}</code>	$\epsilon > 0 \quad (1)$
<code>\begin{eqnarray} f(x) & = & \cos x \\ f'(x) & = & -\sin x \\ \int_0^x f(y) dy & = & \sin x \\ \end{eqnarray}</code>	$f(x) = \cos x \quad (2)$ $f'(x) = -\sin x \quad (3)$ $\int_0^x f(y) dy = \sin x \quad (4)$

Remarques

- Pour L'environnement `eqnarray`, on peut annuler le numéro d'une équation en mettant la commande `\nonumber` à la fin de la ligne concernée
- La commande `\displaystyle` permet, dans un environnement `math` par exemple, de forcer L^AT_EX à utiliser des caractères et des symboles de la même taille qu'en environnement `displaymath`. Cette commande est à utiliser avec modération, car son utilisation risque de modifier l'interligne du paragraphe, pour les formules qui sont placées dans le texte.

12.2 Les polices mathématiques

Commande	Aperçu	Commentaire
<code>\mathrm{formule}</code>	ABC + abc123	roman
<code>\mathbf{formule}</code>	ABC + abc123	gras
<code>\mathsf{formule}</code>	ABC + abc123	sans serif
<code>\mathnormal{formule}</code>	<i>ABC + abc123</i>	<i>normal</i>
<code>\mathbb{formule}</code>	ABC	alphabet <i>Blackboard</i>
<code>\mathcal{formule}</code>	ABC	alphabet calligraphié
<code>\mathfrak{formule}</code>	ABCabc123	alphabet Euler-Œraktur
<code>\mathtt{formule}</code>	ABCabc123	alphabet machine à écrire
<code>\mathit{formule}</code>	<i>ABCabc123</i>	alphabet <i>italique</i>

12.3 Les règles

1. Les espaces sont déduits de la logique de la formule ou indiqués à l'aide de commandes spécifiques telles que `\,`, `\quad`, `\qquad`

Source	Aperçu
<pre>\begin{equation} \forall x \in \mathbb{R} : \qquad x^2 \geq 0 > 0 \end{equation}</pre>	$\forall x \in \mathbb{R} : \quad x^2 \geq 0 \quad (5)$

2. Les lignes vides ne sont pas autorisées. Un seul paragraphe par formule.
3. Une lettre est prise comme le nom d'une variable. Pour insérer du texte normal, il faut utiliser la commande `\textrm`

Source	Aperçu
<pre>\begin{equation} x^2 \geq 0 \quad \qquad \textrm{ pour tout } x \in \mathbb{R} \end{equation}</pre>	$x^2 \geq 0 \quad \text{pour tout } x \in \mathbb{R} \quad (6)$

Remarque 19 La négation des symboles s'obtient en ajoutant `\not` devant l'expression. Par exemple, $\not\leq$ s'obtient avec `\not\leq`.

12.4 Éléments d'une formule

Une liste exhaustive de tous les symboles est disponible en fin de document.

- L'alphabet grec

`\alpha`, `\beta`, `\gamma` α, β, γ

- Les indices `_` et exposants `^`

Source	Aperçu
$\$a_{1}\$ \quad \$x^{2}\$ \quad \$e^{-\alpha t}\$ \quad \$a^3_{ij}\$ \quad \$e^{x^2} \neq e^{x^2}\$$	$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$ $e^{x^2} \neq e^{x^2}$

- Les racines : `\sqrt` `\sqrt[n]` `\surd`

Source	Aperçu
$\$\sqrt{x}\$ \quad \sqrt{x^2+\sqrt{y}} \quad \sqrt[3]{2} \quad \sqrt{x^2+y^2}$	$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$ $\sqrt{x^2 + y^2}$

- Les fonctions en lettres droites

Source	Aperçu
$\$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1\$\$$	$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$

- Les symboles à taille variable

Source	Aperçu
$\$\sum_{i=1}^n \prod_{1 \leq j \leq m} \int_0^{\frac{\pi}{2}} \oint_{\partial\Omega} \bigcap_{i=1}^{\infty} \bigcup_{\alpha}$	$\sum_{i=1}^n \quad \prod_{1 \leq j \leq m} \quad \int_0^{\frac{\pi}{2}}$ $\oint_{\partial\Omega} \quad \bigcap_{i=1}^{\infty} \quad \bigcup_{\alpha}$

- Les délimiteurs

Les délimiteurs peuvent encadrer une expression à l'aide de `\left délimiteur` et `\right délimiteur`. Le délimiteur "." sert alors de délimiteur vide. Ils peuvent être précédés de `\big`, `\Big`, `\bigg`, et `\Bigg` qui modifient leur taille.

Source	Aperçu
$\$1+(\frac{1}{1-x^2})^3\$\$ \quad \$1+\left(\frac{1}{1-x^2}\right)^3\$\$$	$1 + \left(\frac{1}{1-x^2}\right)^3$ $1 + \left(\frac{1}{1-x^2}\right)^3$

- L'environnement `array` pour les matrices et les alignements verticaux

Source	Aperçu
<pre> $X = \left(\begin{array}{ccc} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{array} \right)$ </pre>	$X = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$
<pre> $y = \begin{cases} a & \text{si } d > c \\ b+x & \text{si } d \leq -c \\ l & \text{sinon} \end{cases}$ </pre>	$y = \begin{cases} a & \text{si } d > c \\ b+x & \text{si } d \leq -c \\ l & \text{sinon} \end{cases}$

13 Utilisation des packages

L'objectif de cette section est de mettre en évidence les commandes les plus utiles des packages optionnels que j'ai cités. Il faut savoir qu'il existe bien d'autres commandes. Le lecteur curieux pourra consulter les documentations complètes sur le site du *Loria*.

13.1 Comment installer les packages optionnels

Une fois que vous vous êtes procuré un package optionnel, il faut que vous le placiez à l'emplacement suivant sous *Windows*

```
C:\Program Files\MikTeX\TeX\LaTeX
```

Il faut ensuite que vous mettiez à jour la base de données de *MikTeX* (inventaire des fichiers *sty*) en cliquant dans la barre de menus sur

```
Démarrer\Programmes\MikTeX\MikTeX Options
```

puis sur *Refresh Now* dans l'onglet *General*.

Remarque 20 *Si vous n'avez pas de fichier sty dans le répertoire du nouveau package, il faudra compiler avec \LaTeX un fichier ins pour en générer un. Pour cela, vous pouvez le faire directement avec une commande sous DOS en lançant dans le répertoire approprié latex fichier.ins ou en ouvrant le fichier ins avec l'éditeur et en cliquant sur l'icône de compilation \LaTeX .*

13.2 L'extension *color*

(Package fourni avec MikTeX)

Il faut mettre la ligne de commande `\usepackage{color}` dans le préambule pour disposer des commandes suivantes

<code>\textcolor{couleur}{texte}</code>	change la couleur du texte
<code>\color{couleur}</code>	change la couleur du bloc courant
<code>\colorbox{couleur}{texte}</code>	crée une boîte de couleur
<code>\fcolorbox{bordure}{couleur}{texte}</code>	crée une boîte de couleur encadrée

`\definecolor{nom}{modèle}{valeur}` permet de définir une nouvelle couleur. *modèle* peut être **rgb**, **gray** ou **cmymk**. Les couleurs sont données sous forme de coordonnées de 0 à 1, séparées par des virgules (exemple : `\definecolor{lightgreen}{rgb}{.5,1,.5}`). Les couleurs **black**, **white**, **red**, **green**, **blue**, **yellow**, **cyan** et **magenta** sont définies par défaut.

Remarque 21 *Attention si vous faites une boîte colorée : le texte reste écrit en noir.*

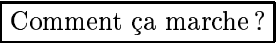
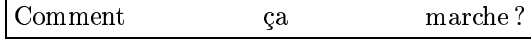
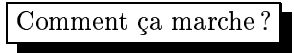
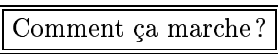
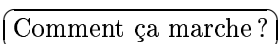
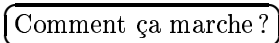

13.3 L'extension *fancybox*

(Package optionnel)

Pour pouvoir utiliser les cadres ou boîtes de \LaTeX il faut mettre la ligne de commande `\usepackage{fancybox}` dans le préambule. On dispose alors des commandes suivantes

13.3.1 Les boîtes de base

Il existe plusieurs genres de boîtes, également appelées cadres

- `\mbox{texte}` : Comment ça marche ?
- `\makebox[taille][position]{texte}` : Comment ça marche ?
- `\fbox{texte}` : 
- `\framebox[taille][position]{texte}` : 
- `\shadowbox{texte}` : 
- `\doublebox{texte}` : 
- `\ovalbox{texte}` : 
- `\Ovalbox{texte}` : 
- `\colorbox{couleur}{texte}` : 

où *taille* est une longueur définie avec les unités usuelles (elle peut également être `\textwidth` si vous voulez que la largeur de la boîte soit celle du texte que vous y insérez), et où *position* peut prendre les valeurs suivantes

position	signification
<i>l</i>	texte aligné à gauche
<i>c</i>	texte centré
<i>r</i>	texte aligné à droite
<i>s</i>	texte étiré sur la longueur de la cellule

13.3.2 Les boîtes avec environnements

Si on veut utiliser une boîte contenant des environnements, on peut utiliser l'architecture suivante. On définit dans un premier temps ce que sera *Sbox*

```

\begin{Sbox}
\begin{minipage}[options]{longueur l}
...
  \begin{env1}
    texte
  \end{env1}
...
  \begin{env2}
    texte
  \end{env2}
...
\end{minipage}
\end{Sbox}

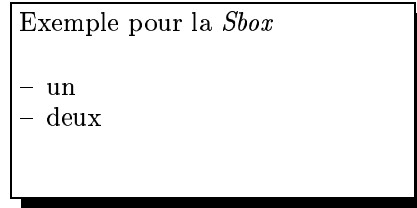
```


puis on l'appelle dans la boîte que l'on veut avec `\TheSbox` en argument.

Exemple

```
\begin{Sbox}
\begin{minipage}{5cm}
Exemple pour la Sbox
\begin{itemize}
\item un
\item deux
\end{itemize}
\end{minipage}
\end{Sbox}
\shadowbox{\TheSbox}
```

affichera



13.4 L'extension *fancyhdr*

(Package optionnel)

Pour pouvoir créer ses propres en-têtes et pieds de page, il faut mettre les lignes de commande suivantes dans le préambule

- `\usepackage{fancyhdr}`
- `\pagestyle{fancy}`, ou `\thispagestyle{fancy}` dans le corps de document pour travailler par page.

Le style de page *fancy* permet d'obtenir un schéma de page avec les en-têtes et pieds suivants

HautGauche	HautCentre	HautDroit
page		
BasGauche	BasCentre	BasDroit

où l'on peut définir chacun des six champs, et de manière différente pour obtenir une mise en page harmonieuse suivant les pages paires ou impaires.

Par défaut dans la mise en page *fancy*, deux lignes d'en-têtes et de pieds de pages sont présentes. Elle ont une largeur de 0.4pt. Si on veut modifier cette épaisseur ou même retirer ces lignes (épaisseur nulle), il faut insérer les lignes de commande suivante dans le préambule

- `\renewcommand{\headrulewidth}{0.2pt}` pour une ligne plus fine en haut
- `\renewcommand{\footrulewidth}{0pt}` pour ne pas avoir de ligne de bas de page.

Pour modifier les champs, il faut utiliser les commandes suivantes dans le préambule

- `\fancyhead[position]{texte}` pour modifier les champs d'en-tête de page
- `\fancyfoot[position]{texte}` pour modifier les champs de bas de page.

où

position	signification
o	page impaire (pour la classe de document <i>book</i>)
e	page paire (pour la classe de document <i>book</i>)
l	champ gauche
c	champ central
r	champ droit

Le *texte* est celui que vous entrez, mais vous pouvez également faire des références aux chapitres ou à la pagination avec les commandes suivantes

texte	signification
<code>\thepage</code>	renvoie le numéro de la page courante
<code>\leftmark</code>	nom de la première partie dans la hiérarchie du document (cf Remarque 1)
<code>\rightmark</code>	nom de la deuxième partie dans la hiérarchie du document (cf Remarque 2)

Remarque 22 1 *L'ordre hiérarchique des parties définies dans le document est celui que vous avez défini. Il correspond à celui donné dans le tableau page ? Par exemple, si vous faites un document de classe *book* ou report, et si vous ne voulez au plus que des chapitres, alors les premières parties dans la hiérarchie sont les chapitres, les secondes parties sont les sections... Par contre, si vous faites un document de classes *article*, alors les premières parties dans la hiérarchie sont les sections, les secondes parties sont les sous-sections...*

Remarque 23 2 *Comme je l'ai indiqué lors de la section ? page ?, il faut distinguer la mise en page selon la classe de document (commandes `\fancyhead` et `\fancyfoot`). Je vais illustrer cela par deux exemples.*

Exemples

- Exemple de pagination pour le document que vous lisez (classe de document *article*)

```

\pagestyle{fancy}
\fancyhead{}
\fancyfoot{}
\renewcommand{\headrulewidth}{.4pt}
\renewcommand{\footrulewidth}{.4pt}
\fancyhead[l]{\textsl{Introduction à LATEX}}
\fancyhead[r]{\leftmark}
\fancyfoot[l]{\textsl{MATEO Damien}}
\fancyfoot[r]{\thepage}

```

Introduction à L ^A T _E X	7 LES COMMANDES
page	
MATEO Damien	Page 6

- Exemple de pagination pour un livre (classe de document *book*)

```

\pagestyle{fancy}
\fancyhead{}
\fancyfoot{}
\renewcommand{\headrulewidth}{.4pt}
\renewcommand{\footrulewidth}{.4pt}
\fancyhead[el,or]{\textsl{Introduction à LATEX}}
\fancyhead[er,ol]{\leftmark}
\fancyfoot[er,ol]{\textsl{ MATEO Damien}}
\fancyfoot[el,or]{\thepage}

```

Introduction à L ^A T _E X	7 LES COMMANDES
page	
Page 6	MATEO Damien

7 LES COMMANDES	Introduction à L ^A T _E X
page	
MATEO Damien	Page 7

Remarque 24 On peut utiliser les commandes `\MakeUppercase` et `\nouppercase` pour passer les titres de chapitres et sections et majuscules ou minuscules.

13.5 L'extension *fancyvrb*

(Package optionnel)

Ce package est une extension de l'environnement *verbatim* avec des options (possibilité de changer les polices, d'ajouter un cadre, des labels...). Il suffit d'insérer la ligne `\usepackage{fancyvrb}` dans le préambule. L'utilisation est la suivante

```

\begin{Verbatim}[options]
  texte
\end{Verbatim}

```

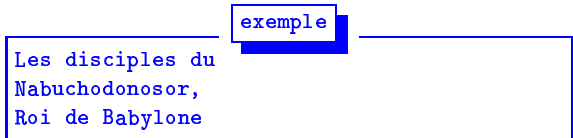
avec un V majuscule !

où

options	exemple de valeurs	signification
<i>fontfamily</i>	<i>helvetica, courier</i>	change la police
<i>fontsize</i>	<code>\small, \scriptsize...</code>	change la taille de la police
<i>fontshape</i>	<i>sl</i>	écrit penché
<i>fontseries</i>	<i>b</i>	pour passer en gras
<i>formatcom</i>	<code>\color{red}</code>	change la couleur d'écriture en rouge (texte, cadre...)
<i>frame</i>	<i>topline</i> (haut) <i>bottomline</i> (bas) <i>lines</i> (haut et bas) <i>single</i> (cadre simple) <i>none</i> (par défaut)	ajoute un cadre autour du texte
<i>framerule</i>	1mm, 2mm ...	épaisseur du cadre
<i>framesep</i>	5mm, 1cm ...	centrement vertical du texte
<i>rulecolor</i>	<code>\color{red}</code>	change la couleur du cadre
<i>numbers</i>	<i>left, right</i>	place un compteur de lignes en dehors du cadre
<i>numbersep</i>	2pt ...	espace entre le compteur de lignes et le texte
<i>stepnumber</i>	2, 5 ...	change le pas pour le compteur de lignes
<i>firstnumber</i>	100	commence à afficher le texte à la 100ème ligne
<i>lastline</i>	100	affiche le texte jusqu'à la 100ème ligne
<i>xleftmargin</i>	5mm	laisse une marge de 5mm à gauche
<i>xrightmargin</i>	5mm	laisse une marge de 5mm à droite
<i>labelposition</i>	<i>topline</i> , ou <i>bottomline</i>	place un titre pour le cadre sur la ligne du haut ou du bas
<i>label</i>	<i>toto.tex</i>	intitulé du cadre du sous forme de texte ou dans une boîte

Exemple

```
\begin{Verbatim}[fontsize=\small,
  frame=single, formatcom=\color{\blue},
  numbers=right, labelposition=topline,
  label=\shadowbox{exemple}]
Les disciples du
Nabuchodonosor,
Roi de Babylone
\end{Verbatim}
```



Remarque 25 Avec le mêmes options, on peut également utiliser les commandes suivantes

- `| VerbatimInput[options]{fichier}` qui insère le contenu de fichier en Verbatim avec toutes les options spécifiées
- `| Define VerbatimEnvironment{env}{Verbatim}{options}` qui définit un nouvel environnement env de type Verbatim avec toutes les options spécifiées. A insérer dans le préambule.

13.6 L'extension *listings*

(Package optionnel)

Cette extension permet d'insérer des codes sources en mettant en évidence les commandes spécifiques au langage, avec un compteur de lignes. Il faut insérer la ligne `\usepackage{listings}` dans le préambule.

L'utilisation est la suivante

```
\begin{lstlisting}{options}
code
\end{lstlisting}
```

où

options	exemple de valeurs	signification
<i>basicstyle</i>	<code>\small</code> , <code>\scriptsize...</code>	change la taille de la police
<i>labelstyle</i>	<code>\tiny...</code>	change la taille des numéros du compteur de lignes
<i>labelstep</i>	2, 3...	change le pas du compteur de lignes
<i>first</i>	100...	commence à afficher le texte à la 100ème ligne
<i>last</i>	100...	affiche le texte jusqu'à la 100ème ligne
<i>frame</i>	<i>t</i> (haut simple) <i>b</i> (bas simple) <i>l</i> (gauche simple) <i>r</i> (droite simple) <i>T</i> (haut double) <i>B</i> (bas double) <i>L</i> (gauche double) <i>R</i> (droite double)	ajoute un cadre autour du texte
<i>blankstring</i>	<i>true</i> , <i>false</i>	affiche ou pas les blancs des sources
<i>language</i>	<i>[ANSI]C</i> <i>[ANSI]C++</i> <i>[77]Fortran</i> <i>[90]Fortran</i> <i>Matlab</i> <i>[XSC]Pascal</i> <i>[LaTeX]Tex</i>	sélectionne le langage mis en évidence

Exemple

```
\begin{lstlisting}{language=Matlab, frame=TBlr,
basicstyle=\scriptsize
labelstyle=\tiny, labelstep=2}
```

```
function C=pmM(A,B)
[n1,m1]=size(A);
[n2,m2]=size(B);
if (m1~=n2)
    disp('Problème de dimension');
    break
end;
for i=1:n1
    for j=1:n2
        S=0;
        for k=1:m1
            S=A(i,k)*B(k,j)+S;
        end
        C(i,j)=S;
    end
end
end
```

```
function C=pmM(A,B)
[n1,m1]=size(A);
[n2,m2]=size(B);
if (m1~=n2)
    disp('Problème de dimension');
    break
end;
for i=1:n1
    for j=1:n2
        S=0;
        for k=1:m1
            S=A(i,k)*B(k,j)+S;
        end
        C(i,j)=S;
    end
end
```

```
\end{lstlisting}
```

Remarques

- Si on est lassé de répéter les options, on peut les définir une fois pour toutes avec la commande `\lstset{options}`, et on appelle le code par

```
\begin{lstlisting}{}
code
\end{lstlisting}
```

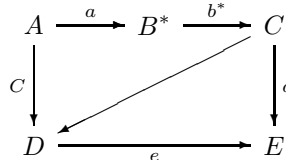
- `\lstinputlisting[options]{fichier}` insère le contenu de *fichier* avec toutes les *options* spécifiées
- Problème de reconnaissance des accents.

13.7 L'extension *pb-diagram*

Cette extension est très utile si on veut faire des schémas. Il faut insérer la ligne `\usepackage{pb-diagram}`. Pour plus de simplicité, je présente l'utilisation de cette extension par des exemples

Exemple

```
$$
\begin{diagram}
\node{A}\arrow[e,t]{a}\arrow{s,l}{C}\arrow{ex}
\node{B^*}\arrow{e,t}{b^*}
\node{C}\arrow{s,r}{d}\arrow{wsw}
\node{D}\arrow[2]{e,b}{e}\node[2]{E}
\end{diagram}
$$
```



Voici les deux commandes principales

- `\node[ncols]{formule}` place une lettre ou un symbole entre des flèches
- `\arrow[taille]{direction, option}`
 ou bien `\arrow[taille]{direction, conf1label, options}{label1}`
 ou bien `\arrow[taille]{direction, conf2label, options}{label1}{label2}`

où

<i>ncols et taille</i>	nombre de colonnes de décalage par rapport à la position courante
<i>formule</i>	lettre, mot ou symbole à mettre entre les flèches
<i>direction</i>	succession de lettres <i>n, e, s, w</i> et tous les dérivés (<i>sswww</i> par exemple)
<i>conf1label</i>	une des quatre lettres <i>t, b, l, r</i> pour placer le label de la flèche
<i>conf2label</i>	<i>tb, bt, lr, rl</i> pour placer deux labels sur une flèche
<i>options</i>	<hr/> .. pour faire une flèche en pointillés - pour ne pas avoir de tête de flèche <> pour avoir deux têtes de flèche 1 pour placer le label au $\frac{1}{4}$ de la flèche 2 pour placer le label au $\frac{1}{2}$ de la flèche (défaut) 3 pour placer le label aux $\frac{3}{4}$ de la flèche

14 Tableaux de commandes

TAB. 1 – Les accents et caractères spéciaux

ó	\'{}o}	ô	\~{}o}	ò	\.{}o}	õ	\~{}o}	â	\aa	æ	\ae	ß	\ss
ò	\'{}o}	ö	\v{}o}	ø	\d{}o}	ø	\c{}o}	Å	\AA	Æ	\AE	¿	?'
ö	\"{}o}	ö	\u{}o}	ö	\={}o}	ø	\o	ł	\l	œ	\oe	¡	!'
ö	\H{}o}	ô	\t{}o}	ø	\b{}o}	Ø	\O	Ł	\L	Œ	\OE		
†	\dag	¶	\P	#	\#	\	\$_backslash\$	{	\{	§	\S	£	\pounds
‡	\ddag	&	\&	\$	\\$	©	\$_copyright\$	}	\}	_	_	%	\%

TAB. 2 – Les accents mathématiques

\hat{a}	\hat{}a}	\check{a}	\check{}a}	\tilde{a}	\tilde{}a}	\acute{a}	\acute{}a}
\grave{a}	\grave{}a}	\dot{a}	\dot{}a}	\ddot{a}	\ddot{}a}	\breve{a}	\breve{}a}
\bar{a}	\bar{}a}	\vec{a}	\vec{}a}	\widehat{A}	\widehat{}A}	\widetilde{A}	\widetilde{}A}

TAB. 3 – Les fonctions mathématiques

\arccos	\cos	\csc	\exp	\ker	\limsup	\min	\sinh
\arcsin	\cosh	\deg	\gcd	\lg	\ln	\Pr	\sup
\arctan	\cot	\det	\hom	\lim	\log	\sec	\tan
\arg	\coth	\dim	\inf	\liminf	\max	\sin	\tanh

TAB. 4 – Divers mathématiques

\overleftarrow{abc}	\overleftarrow{}{abc}	\overrightarrow{abc}	\overrightarrow{}{abc}
\overline{abc}	\overline{}{abc}	\underline{abc}	\underline{}{abc}
$\frac{abc}{xyz}$	\stackrel{}{abc}{xyz}	$\frac{abc}{xyz}$	\frac{}{abc}{xyz}
(mod b)	\pmod{}{b}		

TAB. 5 – Les lettres grecques

α	\alpha	β	\beta	γ	\gamma	δ	\delta
ϵ	\epsilon	ε	\varepsilon	ζ	\zeta	η	\eta
θ	\theta	ϑ	\vartheta	ι	\iota	κ	\kappa
λ	\lambda	μ	\mu	ν	\nu	ξ	\xi
π	\pi	ϖ	\varpi	ρ	\rho	ϱ	\varrho
σ	\sigma	ς	\varsigma	τ	\tau	υ	\upsilon
ϕ	\phi	φ	\varphi	χ	\chi	ψ	\psi
ω	\omega	Γ	\Gamma	Δ	\Delta	Θ	\Theta
Λ	\Lambda	Ξ	\Xi	Π	\Pi	Σ	\Sigma
Υ	\Upsilon	Φ	\Phi	Ψ	\Psi	Ω	\Omega

TAB. 6 – Les symboles à taille variable

\sum	\sum	\prod	\prod	\coprod	\coprod	\int	\int	\oint	\oint
\bigcap	\bigcap	\bigcup	\bigcup	\bigsqcup	\bigsqcup	\bigvee	\bigvee	\bigwedge	\bigwedge
\bigodot	\bigodot	\bigotimes	\bigotimes	\bigoplus	\bigoplus	\biguplus	\biguplus		

TAB. 7 – Les délimiteurs mathématiques

(()	↑	<code>\uparrow</code>	↑	<code>\Uparrow</code>
[[ou <code>\lbrack</code>] ou <code>\rbrack</code>	↓	<code>\downarrow</code>	↓	<code>\Downarrow</code>
{	{ ou <code>\lbrace</code>	} ou <code>\rbrace</code>	↕	<code>\updownarrow</code>	↕	<code>\Updownarrow</code>
<	<code>\langle</code>	>		ou <code>\vert</code>		ou <code>\Vert</code>
⌊	<code>\lfloor</code>	⌋	⌈	<code>\lceil</code>	⌉	<code>\rceil</code>

TAB. 8 – Les symboles mathématiques

±	<code>\pm</code>	∓	<code>\mp</code>	×	<code>\times</code>	÷	<code>\div</code>
*	<code>\ast</code>	★	<code>\star</code>	◦	<code>\circ</code>	•	<code>\bullet</code>
⋅	<code>\cdot</code>	∩	<code>\cap</code>	∪	<code>\cup</code>	⊕	<code>\uplus</code>
⊆	<code>\sqcap</code>	⊆	<code>\sqcup</code>	∨	<code>\vee</code>	∧	<code>\wedge</code>
−	<code>\setminus</code>	∩	<code>\wr</code>	◇	<code>\diamond</code>	▽	<code>\bigtriangledown</code>
△	<code>\lhd</code>	△	<code>\bigtriangleup</code>	▷	<code>\rhd</code>	◁	<code>\triangleleft</code>
∇	<code>\unlhd</code>	▷	<code>\triangleright</code>	▷	<code>\unrhd</code>	⊕	<code>\oplus</code>
⊖	<code>\ominus</code>	⊗	<code>\otimes</code>	⊗	<code>\oslash</code>	⊙	<code>\odot</code>
○	<code>\bigcirc</code>	†	<code>\dagger</code>	‡	<code>\ddagger</code>	∏	<code>\amalg</code>

TAB. 9 – Les symboles de relation

≤	<code>\leq</code>	≥	<code>\geq</code>	≲	<code>\prec</code>	≳	<code>\succ</code>	⪯	<code>\preceq</code>
⪯	<code>\succeq</code>	≪	<code>\ll</code>	≫	<code>\gg</code>	⊆	<code>\subset</code>	⊇	<code>\supset</code>
⊆	<code>\subseteq</code>	⊇	<code>\supseteq</code>	⊆	<code>\sqsubset</code>	⊆	<code>\sqsupset</code>	⊆	<code>\sqsubseteq</code>
⊆	<code>\sqsupseteq</code>	∈	<code>\in</code>	∋	<code>\ni</code>	⊥	<code>\vdash</code>	⊥	<code>\dashv</code>
≡	<code>\equiv</code>	≈	<code>\sim</code>	≈	<code>\simeq</code>	∞	<code>\asymp</code>	≈	<code>\approx</code>
⊥	<code>\perp</code>	≠	<code>\neq</code>	≐	<code>\doteq</code>	∝	<code>\propto</code>	⊥	<code>\models</code>
⊥	<code>\perp</code>	—	<code>\mid</code>		<code>\parallel</code>	⊗	<code>\bowtie</code>	×	<code>\Join</code>
(<code>\smile</code>)	<code>\frown</code>						

TAB. 10 – Les flèches

←	<code>\leftarrow</code>	→	<code>\rightarrow</code>	←	<code>\longleftarrow</code>
→	<code>\longrightarrow</code>	←	<code>\Leftarrow</code>	⇒	<code>\Rightarrow</code>
⇐	<code>\Leftrightarrow</code>	⇒	<code>\Longrightarrow</code>	↔	<code>\leftrightarrow</code>
↔	<code>\longleftrightarrow</code>	⇌	<code>\Leftrightarrow</code>	↔	<code>\Longleftrightarrow</code>
↦	<code>\mapsto</code>	↦	<code>\longmapsto</code>	↵	<code>\hookleftarrow</code>
↵	<code>\hookrightarrow</code>	↵	<code>\leftharpoonup</code>	↵	<code>\rightharpoonup</code>
↶	<code>\leftharpoondown</code>	↶	<code>\rightharpoondown</code>	⇌	<code>\leftrightharpoons</code>
↷	<code>\leadsto</code>	↑	<code>\uparrow</code>	↑	<code>\Uparrow</code>
↓	<code>\downarrow</code>	↓	<code>\Downarrow</code>	↕	<code>\updownarrow</code>
↕	<code>\Updownarrow</code>	↗	<code>\nearrow</code>	↘	<code>\searrow</code>
↘	<code>\swarrow</code>	↙	<code>\nwarrow</code>		

TAB. 11 – Divers symboles

ℵ	<code>\aleph</code>	ℏ	<code>\hbar</code>	ℐ	<code>\imath</code>	ℐ	<code>\jmath</code>	ℓ	<code>\ell</code>
∅	<code>\wp</code>	ℜ	<code>\Re</code>	ℑ	<code>\Im</code>	∪	<code>\mho</code>	′	<code>\prime</code>
∅	<code>\emptyset</code>	∇	<code>\nabla</code>	√	<code>\surd</code>	⊤	<code>\top</code>	⊥	<code>\bot</code>
	<code>\ </code>	∠	<code>\angle</code>	∀	<code>\forall</code>	∃	<code>\exists</code>	△	<code>\triangle</code>
♭	<code>\flat</code>	ℵ	<code>\natural</code>	#	<code>\sharp</code>	\	<code>\backslash</code>	∂	<code>\partial</code>
∞	<code>\infty</code>	□	<code>\Box</code>	◇	<code>\Diamond</code>	♠	<code>\spadesuit</code>	♣	<code>\clubsuit</code>
◇	<code>\diamondsuit</code>	♥	<code>\heartsuit</code>						