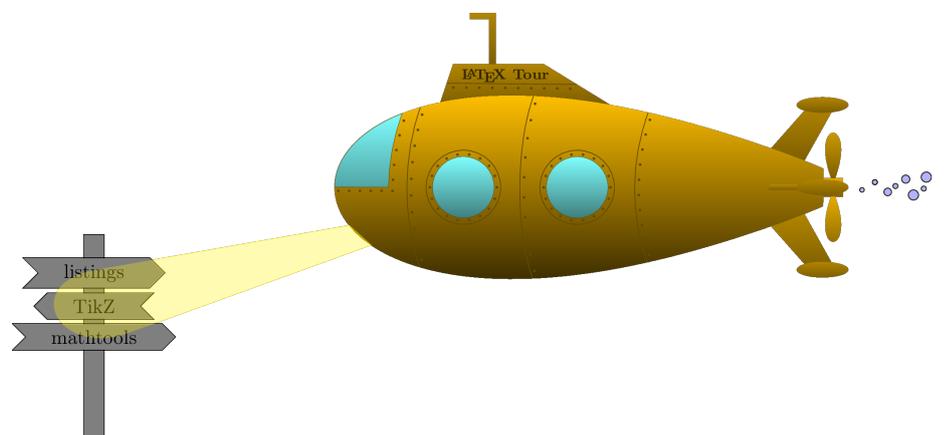


Immersion dans L^AT_EX (Première édition)

Documents en L^AT_EX

NICOLAS KIELBASIEWICZ

22 janvier 2020



1	Introduction	5
2	Installation et utilisation	7
2.1	Installer \LaTeX	7
2.2	Quel éditeur de texte choisir?	7
3	Premier aperçu de \LaTeX	9
3.1	Un exemple pour commencer	9
3.2	Fichiers générés à la compilation	10
3.3	Comprendre les messages d'erreur	10
4	Organisation générale d'un document \LaTeX	12
4.1	La syntaxe de \LaTeX	12
4.2	Classe, préambule, extensions et corps de texte	12
4.3	Quelles extensions utiliser?	13
4.4	Le préambule type	14
5	Structuration d'un document	16
5.1	Page de garde	16
5.2	Chapitrage	17
5.3	La table des matières	17
5.4	Variantes de polices et couleurs	17
5.5	Les listes	19
5.6	Notion de paragraphes et d'espacements	20
6	Les mathématiques	21
6.1	Quel environnement pour les mathématiques?	21
6.2	Opérations usuelles	21
6.3	Variantes de polices	22
6.4	Symboles mathématiques	22
6.5	Lettres grecques	23
6.6	Équations et systèmes d'équations	24
7	Pour aller plus loin . . .	28
7.1	Inclusion d'images	28
7.2	Tableaux et matrices	28
7.3	Utiliser les environnements flottants	30
7.4	Théorèmes	31
A	Préambule \LaTeX de ce document	32
A.1	La liste complète des extensions utilisées	32

1.1	Définition d'une liste numérotée	5
5.2	Commandes et déclarations de changement de style	18
5.3	La commande <code>\emph</code>	19
5.4	Couleurs usuelles	19
5.5	Définition d'une liste non numérotée	19
5.6	Les espacements verticaux	20
6.1	Utilisation des délimiteurs	22
6.2	Ajouter des commentaires avec <code>\overbrace</code> ou <code>\underbrace</code>	24
6.3	L'environnement <i>equation</i>	25
6.4	L'environnement <i>align</i>	25
6.5	Utilisation de <code>\nonumber</code> dans un environnement <i>align</i>	25
6.6	L'environnement <i>aligned</i>	26
6.7	L'environnement <i>cases</i> pour aligner des équations	26
6.8	L'environnement <i>cases</i> pour écrire une alternative	26
7.1	Inclure une image	28
7.2	Notre premier tableau	28
7.3	Tableaux et bordures pleines	29
7.4	Tableaux et bordures partielles	29
7.5	Tableaux mathématiques	30
7.6	Tableau flottant	31

2.1	Fenêtre d'édition de Texworks	7
3.1	Document généré avec le code du tutoriel	9
5.1	En tête d'un document de classe <i>article</i> à gauche et page de garde d'un document de type <i>report</i> à droite	16

6.1	Fonctions et opérateurs usuels	21
6.2	Les lettres grecques minuscules	23
6.3	Les lettres grecques majuscules	23
6.4	Quelques symboles d'opérateurs	23
6.5	Quelques symboles d'opérateurs de taille variable	23
6.6	Quelques symboles de relations	23
6.7	Quelques symboles divers	24
6.8	Quelques flèches	24
6.9	Les accents mathématiques	24
6.10	Les accents mathématiques de taille variable	24
7.1	Exemple de tableau flottant	31

L'histoire commence avec l'invention de $\text{T}_{\text{E}}\text{X}$ en 1977 par le mathématicien et informaticien Donald Knut. Il s'agit d'un système de composition de documents, essentiellement destiné à l'édition de documents techniques. $\text{T}_{\text{E}}\text{X}$ vient du mot grec désignant l'art, la technique : $\tau\acute{\epsilon}\chi\nu\eta$. \LaTeX , quant à lui, est créé en 1983 par le chercheur en informatique Leslie Lamport pour simplifier l'utilisation de $\text{T}_{\text{E}}\text{X}$ ¹.

Il s'agit donc d'un langage dédié à la génération de documents de toutes sortes : articles, rapports, mémoires, livres, lettres, romans, conférences, posters, ... et même partitions musicales!!! En particulier, il offre la possibilité de définir des formules mathématiques extrêmement simplement et avec une typographie qui est la référence aujourd'hui.

De manière générale, pour composer un document, il y a 2 méthodes :

- Utiliser des logiciels « tel écran, tel écrit » (en anglais WYSIWYG, acronyme de « What You See Is What You Get »), comme les logiciels propriétaires Microsoft Word (format .doc ou .docx), Apple Pages (format .pages) ou leurs équivalents dans le monde libre : OpenOffice, LibreOffice, ... (format .odt), et bien d'autres encore ...
- Utiliser des éditeurs de texte « basiques » pour écrire du texte éventuellement en utilisant certaines conventions : fichiers .txt, fichiers .html, fichiers Markdown .md, ... et générer un document dans un ou plusieurs formats spécifiques. On parle de contexte WYSIWYM (acronyme de « What You See is What You Mean », que l'on pourrait traduire par « Ce que vous voyez est ce que vous voulez dire »).

\LaTeX rentre dans cette deuxième catégorie, ce qu'illustre l'exemple suivant :

Exemple 1.1 Définition d'une liste numérotée

<code>\begin{enumerate}</code>	
<code>\item blabla bla</code>	1. blabla bla
<code>\item bla blabla</code>	2. bla blabla
<code>\item bla bla bla</code>	3. bla bla bla
<code>\end{enumerate}</code>	

Si vous n'avez connu que des logiciels WYSIWYG, où la mise en forme de votre document se fait entièrement à la souris, vous serez un peu dérouté par la nécessité de « compiler » votre document pdf pour voir le rendu, et bien entendu, il va falloir le faire régulièrement pour éviter de mauvaises surprises!!! C'est la même méthodologie que l'écriture d'un programme informatique que l'on teste au fur et à mesure de son développement.

D'un autre côté, je vous garantis que lorsque vous aurez rédigé votre premier document en \LaTeX , vous verrez tout de suite la différence de qualité en terme de typographie par rapport à la concurrence. Ce n'est pas un hasard si les logiciels WYSIWYG permettent de définir des boîtes dans lesquelles on écrit des formules mathématiques en \LaTeX .

Par ailleurs, la communauté \LaTeX est très développée et un très grand nombre d'extensions sont disponibles, certaines d'entre elles étant trop restrictives, d'autres étant désuètes. Savoir identifier quelles sont les extensions à utiliser ou à éviter n'est pas toujours facile.

1. \LaTeX signifie Lamport $\text{T}_{\text{E}}\text{X}$

L'extension d'un fichier \LaTeX est `.tex`. C'est un fichier texte dans lequel on manipule des commandes (ou macros) et des environnements. Le document généré est principalement un fichier PDF², le format de document le plus couramment utilisé à l'heure actuelle³.

A titre personnel, j'ai découvert \LaTeX en 2002 à l'ENSTA Paris grâce à Marc Baudoin et sa documentation personnelle qu'il a partagé avec les étudiants. A l'époque, elle en était à sa deuxième version, elle en est aujourd'hui à la cinquième : « Apprends LaTeX (5ème édition) » (voir [2]). Comme tout le monde, j'ai mis un peu de temps à m'y mettre (surtout qu'à l'époque, la compilation directe en PDF était loin d'être le standard et les extensions étaient beaucoup moins nombreuses et moins performantes qu'aujourd'hui), mais aujourd'hui, je ne saurais m'en passer, même pour les documents les plus élémentaires.

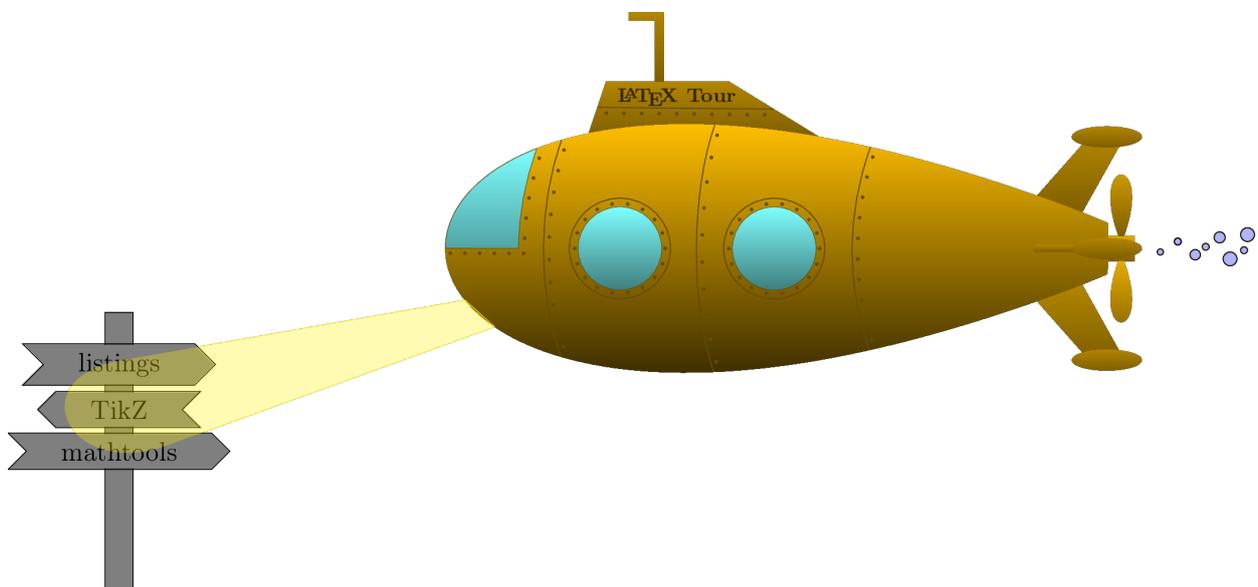
Mis à part la documentation de Marc Baudoin, la référence absolue, bien qu'elle commence à dater (2004), est « The \LaTeX Companion (2nd Edition) » de Frank Mittelbach et Michel Goossens (voir [26]).

Enfin, vous avez :

- le site web de référence, celui du CTAN⁴ <https://www.ctan.org> où vous trouverez la documentation officielle de la plupart des extensions existantes.
- La principale FAQ française <http://www.grappa.univ-lille3.fr/FAQ-LaTeX>
- Une FAQ anglaise très bien fournie <https://texfaq.org>

Dans ce cours introductif à \LaTeX , seules les extensions recommandées et les plus couramment utilisées seront évoquées ou présentées, comme étant le résultat de mon expérience personnelle, afin de vous éviter une recherche fastidieuse sans savoir réellement quoi chercher. Dans la bibliographie de ce document page 36, vous trouverez les liens vers la page CTAN de ces extensions.

Dans cette version abrégée, nous allons vous présenter \LaTeX dans le but de rédiger un document manuscrit : un rapport, une lettre, un livre, un article, ...



2. Mais on peut aussi générer entre autres des fichiers PostScript ou des fichiers HTML

3. PDF est l'acronyme de Portable Document Format, a été créé en 1992 et est devenu une norme ISO en 2008

4. acronyme de Comprehensive \TeX Archive Network

2.1 Installer L^AT_EX

L^AT_EX est un ensemble d'extensions à télécharger, qui fait plusieurs Go!!! On parle de distribution L^AT_EX.

sur Windows La distribution à télécharger s'appelle MikTeX <https://miktex.org>

sur Linux La distribution à télécharger s'appelle TexLive <https://www.tug.org/texlive/>

sur MacOS La distribution à télécharger s'appelle MacTeX <http://www.tug.org/mactex/>. Il s'agit d'une version augmentée de la TeXLive, avec des applications spécifiques à Mac.

2.2 Quel éditeur de texte choisir?

Les éditeurs simples vi, emacs, gedit, kedit, ... (Linux), wordpad (Windows), TextEdit (Mac) ...

Les éditeurs avancés multi-plateformes Visual Studio Code¹, Sublime Text, Atom, ...

Les éditeurs spécifiques OS-dépendants TexShop (Mac)¹, TeXnicCenter (Windows), ...

Les éditeurs spécifiques multi-plateformes Texworks¹ (figure 2.1), TeXstudio, TeXmaker, ...

```

1 \documentclass[11pt,french]{report}
2
3 \usepackage{myPoly}
4
5 % chargement du fichier de biblio
6 \addbibresource{latex.bib}
7
8 \hypersetup{%
9 pdftitle={Initiation à \LaTeX},%
10 pdfauthor={N. Kielbasiewicz},%
11 pdfpagemode=UseOutlines,%
12 }%
13
14 \date{} % date %
15 \title{Initiation à \LaTeX} % titre %
16 \author{Nicolas Kielbasiewicz} % auteur %
17
18 \begin{document}
19
20 \maketitle
21
22 \doparttoc[n]
23 \noptcrule
24 \setcounter{tocdepth}{0}
25 \tableofcontents
26
27 \part{\LaTeX}
28
29 \chapter{Introduction}
30
31 \section{\LaTeX, c'est quoi ?}
32
33 L'histoire commence avec l'invention de \TeX{} en 1977 par le mathématicien et informaticien Donald Knut. Il s'agit d'un système de
composition de documents, essentiellement destiné à l'édition de documents techniques. \TeX{} vient du mot grec désignant l'art,
la technique: $\tau\alpha\upsilon\dot{\epsilon}\pi\sigma\iota\nu$ $\chi\eta\iota$ $\nu\omicron\lambda\epsilon\tau\alpha$. \LaTeX, quant à lui, est créé en 1983 par le chercheur en informatique Leslie
Lamport pour simplifier l'utilisation de \TeX{} \footnote{\LaTeX{} signifie Lamport \TeX{}}.

```

FIGURE 2.1 – Fenêtre d'édition de Texworks

Répondre aux questions suivantes peut vous aider à faire votre choix :

- Souhaitez vous les commandes les plus courantes dans les menus en mode presse-bouton?

1. Les éditeurs que j'ai utilisés à titre personnel

- Travaillez-vous sur plusieurs OS ?
- Souhaitez-vous la visualisation du document généré, réactualisé à chaque compilation ?

Si vous utilisez un éditeur ne disposant pas d'un bouton pour générer le document (soit un éditeur simple, soit un éditeur avancé pour lequel vous n'avez pas chargé de module spécifique pour \LaTeX), il vous faudra alors lancer la génération en ligne de commande, en tapant :

```
pdflatex monfichier.tex
```

Cette possibilité de générer le pdf en ligne de commande peut paraître contraignante pour les néophytes, voire trop « informatique ». Et pourtant, cela ouvre un champ de possibilités extrêmement large. En effet, \LaTeX apporte par exemple des solutions dans des domaines très divers, y compris dans le domaine du web. Ne vous est-il jamais arrivé d'adhérer en ligne à une association ou autre et de récupérer à la validation de votre inscription un document pdf qui contient toutes les informations remplies dans le formulaire ? Et bien, dites vous qu'il est possible (probable ?) que ce document pdf a été généré par \LaTeX !

Il existe enfin un autre moyen de générer des documents \LaTeX : par le biais de plateformes en ligne, comme Overleaf². Vous pouvez créer des documents en ligne, en bénéficiant de templates, mais aussi et surtout de manière collaborative.

2. <https://www.overleaf.com>

3.1 Un exemple pour commencer

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

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[nomath]{kpfonts}
\usepackage{geometry}
\usepackage{babel}
\usepackage{microtype}

% modification des marges %
\geometry{top=2.5cm, bottom=2.5cm, left=1.8cm, right=1.8cm}

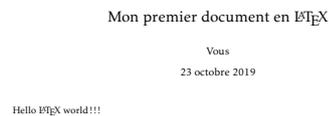
\title{Mon premier document en \LaTeX} % titre %
\author{Vous} % auteur %

\begin{document}

\maketitle

Hello \LaTeX\ world !!!

\end{document}
```



Mon premier document en L^AT_EX

Vous

23 octobre 2019

Hello L^AT_EX world!!!

FIGURE 3.1 – Document généré avec le code du tutoriel

3.2 Fichiers générés à la compilation

La compilation du document `tutorial.tex`, si elle se déroule sans erreurs, générera principalement 3 fichiers : `tutorial.pdf` bien sûr, un fichier auxiliaire `tutorial.aux` contenant principalement les références croisées et le fichier de log `tutorial.log`.

3.3 Comprendre les messages d'erreur

La compilation d'un document \LaTeX génère plusieurs fichiers d'extensions différentes. Le fichier d'extension `.log` contient tout ce qui s'affiche dans le terminal quand vous compilez en ligne de commande, ou dans la console de sortie si vous utilisez un éditeur spécifique.

```
This is pdfTeX, Version 3.14159265-2.6-1.40.20 (TeX Live 2019) (preloaded format=pdflatex
2019.10.23) 23 OCT 2019 11:02
entering extended mode
restricted \write18 enabled.
%&-line parsing enabled.
**tutorial.tex
(./tutorial.tex
LaTeX2e <2019-10-01> patch level 1
```

...

```
8945 multiletter control sequences out of 15000+600000
33971 words of font info for 54 fonts, out of 8000000 for 9000
1141 hyphenation exceptions out of 8191
33i,6n,42p,220b,425s stack positions out of 5000i,500n,10000p,200000b,80000s
{/usr/local/texlive/2019/texmf-dist/fonts/enc/dvips/base/8r.enc}</usr
r/local/texlive/2019/texmf-dist/fonts/type1/public/kpfonts/jkpmn8a.pfb>
Output written on tutorial.pdf (1 page, 12916 bytes).
PDF statistics:
13 PDF objects out of 1000 (max. 8388607)
8 compressed objects within 1 object stream
0 named destinations out of 1000 (max. 500000)
16385 words of extra memory for PDF output out of 17280 (max. 10000000)
```

Je vous épargne les quelques 387 lignes intermédiaires qui se rapportent au chargement des extensions.

Voilà quelques messages d'erreurs potentiels :

Vous utilisez une commande qui n'existe pas

```
! Undefined control sequence.
1.20 Hello \latex
      \ world !!!
?
```

Vous trouvez la ligne de votre document sur laquelle se trouve l'erreur. La commande qui cause l'erreur est suivie d'un saut de ligne. Ici `\latex` qui n'existe pas quand elle est écrite toute en minuscules.

Vous avez ouvert une accolade et oublié de la refermer

```
! Missing } inserted.
<inserted text>
      }
1.22 \end{enumerate}
?
```

Cette erreur est lancée quand l'accolade manquante se trouve à l'intérieur d'un environnement (ici, un *enumerate*). La ligne signalée est celle de clotûre de l'environnement dans lequel l'accolade manque.

Vous avez ouvert un environnement et fermé avec un autre

```
! LaTeX Error: \begin{itemize} on input line 22 ended by \end{description}.
```

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

```
1.24 \end{description}
```

?

Vous pouvez aussi déborder de la zone de texte, auquel cas vous aurez un message du genre :

```
Overfull \hbox (90.97733pt too wide) in paragraph at lines 20--23  
[]$[]$  
[]
```

4.1 La syntaxe de L^AT_EX

Pour L^AT_EX, une commande est un mot composé en règle générale uniquement de lettres et qui commence par un antislash. Par exemple, `\LaTeX`, `\documentclass`, `\usepackage` sont des commandes.

Une commande peut avoir des arguments obligatoires (chaque argument est entre accolades), et/ou des arguments optionnels (l'ensemble des arguments optionnels est entre crochets et chaque argument optionnel est séparé par une virgule à l'intérieur).

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

Par exemple, dans l'exemple ci-dessus, `\documentclass` a un argument, *article* et 3 arguments optionnels, *11pt*, *a4paper* et *french*.



Vous avez sans doute remarqué dans l'exemple de la section 3.1 que la commande `\LaTeX` est suivie d'un anti-slash. La raison est simple, c'est que par défaut il n'y a aucun espacement entre une commande et ce qui la suit. En lieu et place de l'anti-slash, nous aurions pu mettre la commande entre accolades ou lui ajouter un argument vide (donc entre accolades). Un anti-slash suivi d'un espace est une façon d'imposer un espace. Accessoirement, je trouve que c'est la manière la plus lisible pour rajouter un espace suite à une commande.

Il existe des commandes sans arguments qui formatent la suite du corps de texte sans avoir de limite de portée. Il s'agit des déclarations. Pour limiter la portée, on utilise les accolades. Cela donne une syntaxe de la forme :

```
{\cmd bla bla bla}
```

Nous avons également vu une autre notion, celle d'un environnement. Le contenu d'un environnement est encerclé par un `\begin` et par un `\end`, prenant tous deux en argument le nom de l'environnement. *document* et *enumerate* sont les 2 exemples que vous avez rencontrés jusqu'ici.

4.2 Classe, préambule, extensions et corps de texte

Décryptons un peu l'exemple de la section 3.1.

La première ligne d'un document L^AT_EX définit la classe de document :

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

La classe de document définit la mise en page générale du document et les commandes principales de structuration du document. Les classes principales sont :

article pour la rédaction d'articles, c'est-à-dire essentiellement des documents relativement courts sans page de garde, constitués de sections.

report pour la rédaction de rapports ou de mémoires, donc des documents plus longs et constitués de chapitres.

book pour la rédaction de livres, des documents encore plus longs, constitués de chapitres, et destinés à être imprimés en recto-verso et reliés.

beamer pour la rédaction de présentations, composées de diapositives. Il existe d'autres classes, mais *beamer* est de loin la plus complète et la plus utilisée.

tikzposter pour la rédaction de posters, c'est-à-dire un document d'une seule page au format A0. Il existe d'autres classes de qualité équivalente, comme *beamerposter*

Le corps de texte est le contenu effectif du document à rédiger. Il est contenu à l'intérieur d'un environnement *document* :

```
\begin{document}

\maketitle

Hello \LaTeX\ world !!!

\end{document}
```

Le préambule d'un document \LaTeX est tout ce qui se trouve entre la définition de la classe et le début de l'environnement *document*.

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[nomath]{kpfonts}
\usepackage{geometry}
\usepackage{babel}
\usepackage{microtype}

% modification des marges %
\geometry{top=2.5cm, bottom=2.5cm, left=1.8cm, right=1.8cm}

\title{Mon premier document en \LaTeX} % titre %
\author{Vous} % auteur %
```

Il contient 2 types d'éléments :

1. L'inclusion d'extensions (ou packages), avec la commande `\usepackage`. Une extension est un ensemble de commandes et d'environnements dédiés à un besoin particulier, à priori indépendamment de la classe de document.
2. Le paramétrage des extensions et donc du document (définition des couleurs, styles des titres, commandes personnalisées, ...). C'est aussi là que l'on définit le titre, l'auteur et la date du document, qui seront exploités par la commande `\maketitle`



On peut définir ses propres classes de documents et/ou extensions. Il s'agit de fichiers d'extensions respectives `.cls` et `.sty`.

4.3 Quelles extensions utiliser ?

4.3.1 Les extensions systématiques

Voilà les 6 premières extensions qu'il faut inclure :

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[nomath]{kpfonts}
\usepackage{geometry}
\usepackage{babel}
\usepackage{microtype}
```

INPUTENC concerne l'encodage des caractères. Cela doit se conformer à l'encodage du fichier \LaTeX que vous êtes en train d'écrire.

FONTENC concerne l'encodage de la police de caractères. *T1* est l'encodage le plus couramment utilisé. C'est pour dire que l'on utilise des polices de caractères de type 1.

KPFFONTS charge la police de caractères **KPFFONTS**. C'est l'une des polices recommandées avec **LMODERN**, et éventuellement **LIBERTINE** ou **FOURIER**. A noter que l'extension est chargée avec une option, *nomath* qui désactive l'utilisation de la police mathématique de **KPFFONTS** (qui est moins belle que la police mathématique par défaut, mais ce n'est que mon avis).

GEOMETRY définit les marges d'une page, comme montré dans l'exemple en début de chapitre. Voir [7] pour plus de détails.

BABEL charge la langue dans laquelle vous allez écrire. En effet, chaque langue a ses propres règles typographiques et cette extension est là pour les prendre en compte à la génération du document.

MICROTYPE apporte des raffinements d'ordre typographique, comme l'ajustement de l'espacement permettant une meilleure coupure des mots. Le simple fait de charger cette extension suffit à avoir des améliorations notables. Voir [32] pour plus de détails.

4.3.2 Les extensions courantes à utiliser

L'extension GRAPHICX Cette extension concerne l'inclusion des images, et les formats d'images pris en comptes. Voir [4]

L'extension XCOLOR Cette extension concerne la définition des couleurs, avec la possibilité de les mélanger de manière simple. Voir [18]

Les extensions de maths Pour pouvoir écrire des formules mathématiques, il faut charger des extensions de symboles et d'environnements. Les extensions suivantes sont en général auto-suffisantes : **MATHTOOLS** (qui remplace **AMSMATH**, voir [23]), **LATEXSYM** et **AMSSYMB**, pour charger la plupart des symboles, et **AMSTHM** (voir [1]) pour définir et gérer les théorèmes.

L'extension LISTINGS Cette extension est à l'heure actuelle l'extension la plus complète pour insérer des listings de code, avec un nombre élevé de langages pré-définis, et la possibilité de définir ses propres langages pour la mise en page d'un listing. Voir [16].

L'extension HYPERREF Cette extension est dédiée aux références croisées et à la possibilité de mettre des liens cliquables dans des documents. Voir [17].

4.4 Le préambule type

La grande diversité d'extensions fait qu'il existe d'une part des redondances entre certaines d'entre elles, et d'autre part des problèmes de compatibilité. Pour résoudre cela, il faut non seulement choisir les bonnes extensions, mais aussi les charger dans un certain ordre.

Les règles générales sont assez simples :

1. Les extensions **INPUTENC**, **FONTENC** et **KPFFONTS** (ou **LMODERN**) sont à charger en premier.
2. Sauf contre-indication, **BABEL** est la dernière extension à charger, les exceptions les plus courantes étant **HYPERREF**, **LISTINGS** et **MICROTYPE**.
3. Quand une extension doit être chargée avant un autre, elle affiche de plus en plus souvent un warning à la génération du document.

Avec l'ensemble des extensions évoquées jusqu'à présent, le préambule s'écrira :

```
\usepackage[utf 8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[nomath]{kpfonts}
\usepackage{geometry}
\usepackage{graphicx}
\usepackage{xcolor}
\usepackage{mathtools}
\usepackage{latexsym}
\usepackage{amssymb}
\usepackage{amsthm}
\usepackage{tikz}
\usepackage[autocite=inline]{biblatex}
\usepackage{babel}
\usepackage{microtype}
\usepackage{listings}
\usepackage[plainpages=false , pdfpagelabels , bookmarksnumbered , breaklinks=true]{hyperref}
```

Pour aller plus loin et voir la liste des extensions utilisées pour le présent document, voir annexe A.

5.1 Page de garde

Par défaut, 3 informations sont demandées dans le préambule : le titre du document, le ou les auteurs, et la date. Si la date n'est pas précisée, alors celle qui correspond à la date de la dernière compilation du document est utilisée. Ces 3 informations permettent de définir quand elle existe la page de garde du document, ou à défaut l'entête d'un article. L'utilisation de la commande `\maketitle` permet de générer ce titre ou cette page de garde.

```
\title{Mon premier document en \LaTeX} % titre %  
\author{Vous} % auteur %  
  
\begin{document}  
  
\maketitle
```

Mon premier document en \LaTeX
Vous
11 octobre 2019

Mon premier document en \LaTeX
Vous
11 octobre 2019

FIGURE 5.1 – En tête d'un document de classe *article* à gauche et page de garde d'un document de type *report* à droite

5.2 Chapitrage

Il existe 7 niveaux de titres dans un document \LaTeX . Suivant la classe que vous utilisez (`book`, `report`, ou `article`), certains niveaux ne sont pas disponibles.

- `\part`** Définition d'un titre de niveau le plus élevé (disons niveau 1) : la partie. La commande `\part` n'est disponible que pour les classes `book` et `report` et chaque partie est numérotée.
- `\chapter`** Définition d'un titre de niveau 2 : le chapitre. La commande `\chapter` n'est disponible que pour les classes `book` et `report` et chaque chapitre est numéroté.
- `\section`** Définition d'un titre de niveau 3 : la section. C'est le niveau le plus élevé disponible dans la classe `article`. Chaque section est numérotée.
- `\subsection`** Définition d'un titre de niveau 4 : la sous-section, qui est là encore numérotée.
- `\subsubsection`** Définition d'un titre de niveau 5 : la sous-sous-section, qui n'est numérotée que dans la classe `article`.
- `\paragraph`** Définition d'un titre de niveau 6 : le paragraphe. Il n'est pas numéroté. Contrairement aux 5 premiers niveaux, le texte qui suit n'est pas précédé d'un passage automatique à la ligne.
- `\subparagraph`** Définition d'un titre de niveau 7 : le sous-paragraphe. Il n'est pas numéroté et par rapport au `\paragraph`, ce titre est indenté.



Il existe des versions dites étoilées (les commandes sont donc `\part*`, `\chapter*`, ...) qui définissent un titre de niveau correspondant, mais sans numérotation.



On peut également définir une annexe, auquel cas les titres de l'annexe ne sont pas numérotés par des chiffres, mais par des lettres. Il suffit pour cela d'utiliser la commande `\appendix`, sans arguments, pour déclarer que tout ce qui suit est l'annexe du document.

5.3 La table des matières

Maintenant que l'on a défini la structure de notre document, on peut souhaiter avoir une table des matières de notre document.

Pour cela, à l'endroit souhaité, il suffit juste d'appeler la commande `\tableofcontents`, comme on a pu le voir en page 1



On peut également souhaiter avoir une table des matières locales à chaque partie, ou chaque chapitre. Voir pour cela l'extension `MINITOC` [11]



Quand on demande une table des matières, il faut compiler le document 2 fois : une première fois pour générer le fichier auxiliaire d'extension `.aux`, la deuxième fois pour en tenir compte à la génération du document.

5.4 Variantes de polices et couleurs

Nous avons vu jusqu'à présent comment organiser du texte sans réellement agir sur celui-ci. Bien entendu, \LaTeX permet de formater le texte lui-même.

On peut tout d'abord contrôler la taille du texte :

`\tiny` Exemple : Ceci est une phrase de taille *tiny*

`\scriptsize` Exemple : Ceci est une phrase de taille *scriptsize*

`\footnotesize` Exemple : Ceci est une phrase de taille *footnotesize*

`\normalsize` Exemple : Ceci est une phrase de taille *normalsize*

`\large` Exemple : Ceci est une phrase de taille *large*

`\Large` Exemple : Ceci est une phrase de taille *Large*

`\LARGE` Exemple : Ceci est une phrase de taille *LARGE*

`\huge` Exemple : Ceci est une phrase de taille *huge*

`\Huge` Exemple : Ceci est une phrase de taille *Huge*



La commande `\normalsize` ne fait rien, cette taille correspond en effet à la taille définie en option de la classe de document.



Ces commandes sont plutôt destinées à la personnalisation des titres, des tableaux ou des listings de codes qu'à modifier le corps de texte proprement dit (ce qui est désagréable à lire).

On peut également changer le style du texte :

Exemple 5.2 Commandes et déclarations de changement de style

<code>\textit{italique}, {\itshape italique} \\ \textsl{inclin\`e}, {\slshape inclin\`e} \\ \</code>	<i>italique, italique</i> <i>incliné, incliné</i>
<code>\textsc{petites capitales}, {\scshape petites capitales} \\ \textup{droit}, {\upshape droit} \\ \textmd{maigre}, {\mdseries maigre} \\ \textbf{gras}, {\bfseries gras} \\ \textrm{romain}, {\rmfamily romain} \\ \textsf{sans s\`erif}, {\sffamily sans s\`erif} \\ \texttt{machine \`a \`ecrire}, \tffamily machine \`a \`ecrire}</code>	PETITES CAPITALES, PETITES CAPITALES droit, droit maigre, maigre gras, gras romain, romain sans sérif, sans sérif machine à écrire, machine à écrire



On peut bien entendu combiner les commandes et les déclarations de changements de style. Le résultat dépendra néanmoins de ce qu'autorise la police de caractère chargée. C'est aussi pour cette question de combinaison que je préfère, à titre personnel, utiliser les déclarations de changement de style que les commandes. En effet, avec les déclarations, nous n'avons besoin que d'une seule paire d'accollades ! Néanmoins, il existe des situations qui exigent d'utiliser soit les commandes, soit les déclarations.



Un mot sur l'utilisation du texte en italique. On l'utilise en général pour mettre en valeur une partie d'un texte. Il existe en \LaTeX une commande pour mettre en valeur du texte : `\emph` qui est l'exemple typique de la philosophie WYSIWYM de \LaTeX , à savoir différencier la structure du document et son rendu visuel. `\emph` modifie en effet le style du texte en fonction du style de départ :

- Pour du texte normal, `\emph` est l'équivalent de `\textit`, le texte mis en valeur sera en italique.
- Pour du texte qui est déjà en italique, si `\textit` ne fera rien, `\emph` va le mettre en valeur en étant cette fois l'équivalent de `\textup`, à savoir du texte normal.

Exemple 5.3 La commande `\emph`

```
{\upshape Bonjour, \emph{cher utilisateur}, comment allez-vous ?} \\  
{\itshape Bonjour, \emph{cher utilisateur}, comment allez-vous ?}
```

Bonjour, *cher utilisateur*, comment allez-vous?
Bonjour, cher utilisateur, *comment allez-vous?*

On peut enfin changer la couleur du texte à l'aide de la commande `\color` :

Exemple 5.4 Couleurs usuelles

<code>\begin{tabular}{cc}</code>		bleu	rouge
<code>{\color{blue} bleu}</code>	<code>& {\color{red} rouge} \\ </code>	vert	jaune
<code>{\color{green} vert}</code>	<code>& {\color{yellow} jaune} \\ </code>	orange	rose
<code>{\color{orange} orange}</code>	<code>& {\color{pink} rose} \\ </code>	violet	noir
<code>{\color{violet} violet}</code>	<code>& {\color{black} noir} \\ </code>	blanc	gris
<code>{\color{white} blanc}</code>	<code>& {\color{gray} gris} \\ </code>	cyan	magenta
<code>{\color{cyan} cyan}</code>	<code>& {\color{magenta} magenta}</code>		
<code>\end{tabular}</code>			

L'extension `xcolor` vous permet également de définir des couleurs, de les mélanger, ... (voir [18])

5.5 Les listes

Les 3 types de listes ont chacun leur environnement :

`itemize` pour les listes non numérotées. En français, les puces sont des tirets. En anglais, elles sont rondes.

Exemple 5.5 Définition d'une liste non numérotée

<code>\begin{itemize}</code>	• item 1
<code>\item item 1</code>	
<code>\item item 2</code>	• item 2
<code>\item item 3</code>	
<code>\end{itemize}</code>	• item 3

`enumerate` pour les listes numérotées (voir exemple 1.1)

`description` pour les listes descriptives, telles que celle dans laquelle nous sommes pour définir les 3 environnements de type liste.



On peut souhaiter personnaliser les puces de liste comme mettre de la couleur par exemple, ou choisir la puce d'un *itemize*. L'extension `ENUMITEM` est faite pour cela (voir [21]).

5.6 Notion de paragraphes et d'espacements

Dans un document `.tex`, pour changer de paragraphe, il suffit de passer une ligne.

Si maintenant on veut passer une ligne dans le document final, on peut utiliser l'une des 3 commandes d'espacement vertical, à savoir `\smallskip`, `\medskip` et `\bigskip`.

Exemple 5.6 Les espacements verticaux

ligne 0	
ligne 1	
<code>\smallskip</code>	ligne 0
ligne 2	ligne 1
<code>\medskip</code>	ligne 2
ligne 3	ligne 3
<code>\bigskip</code>	ligne4
ligne4	

Par défaut, le texte est justifié (aligné à gauche et à droite simultanément). On peut utiliser les environnements `center`, `flushleft` et `flushright` pour centrer le texte ou ne l'aligner que d'un côté.



Il existe une déclaration équivalente pour le centrage du texte : `\centering`. Il est fortement conseillé de l'utiliser à l'intérieur d'autres environnements si vous voulez centrer le contenu, comme c'est le cas des environnements flottants (voir la section 7.3 les concernant).

6.1 Quel environnement pour les mathématiques ?

Pour écrire une formule mathématique, on utilisera l'un des 2 environnements suivants :

math pour des formules insérées dans le corps de texte. Démarrer et terminer la formule mathématique par le symbole \$. Il s'agit de l'environnement (ou mode) *math*, permettant d'écrire des formules mathématiques dans le corps de texte.

displaymath Démarrer la formule mathématique par `\[` et la terminer par `\]`. Il s'agit de l'environnement (ou mode) *displaymath* et dans ce cas, la formule est seule sur la ligne.



La police de caractère utilisée pour les formules mathématiques est différente de la police du texte. Si l'on souhaite écrire du texte avec la police du texte standard à l'intérieur des environnements *math* et *displaymath*, on utilisera la commande `\text` (ou `\mbox`).

6.2 Opérations usuelles

6.2.1 Les fractions

Pour écrire une fraction, on utilise la commande `\frac` prenant en arguments le numérateur, puis le dénominateur.

Exemple de `\frac{x+1}{y-1}` en mode *displaymath*, puis en mode *math* :

$$\frac{x+1}{y-1} \quad \frac{x+1}{y-1}$$

6.2.2 Exposants et indices

Pour définir un indice, on utilise le caractère `_` et pour définir un exposant, on utilise le caractère `^`. Si l'exposant ou l'indice comporte 2 caractères ou plus, il faudra utiliser des accolades. De même, on peut utiliser des accolades pour indiquer ce sur quoi s'applique l'indice ou l'exposant.

Exemple de `{(x_1+y_{12})}^2=x_1^2+2xy_{12}+{y_{12}}^2` : $(x_1 + y_{12})^2 = x_1^2 + 2xy_{12} + y_{12}^2$

6.2.3 Fonctions et opérateurs usuels et délimiteurs

LaTeX fournit la plupart des symboles de fonctions et opérateurs usuels, comme les fonctions trigonométriques, les fonctions logarithmiques, ... :

cos	<code>\cos</code>	cosh	<code>\cosh</code>	sin	<code>\sin</code>	sinh	<code>\sinh</code>	tan	<code>\tan</code>	tanh	<code>\tanh</code>
arccos	<code>\arccos</code>	arcsin	<code>\arcsin</code>	arctan	<code>\arctan</code>	cot	<code>\cot</code>	coth	<code>\coth</code>		
exp	<code>\exp</code>	ln	<code>\ln</code>	log	<code>\log</code>						
arg	<code>\arg</code>	gcd	<code>\gcd</code>	det	<code>\det</code>	dim	<code>\dim</code>	lim	<code>\lim</code>		
inf	<code>\inf</code>	sup	<code>\sup</code>	min	<code>\min</code>	max	<code>\max</code>				

TABLE 6.1 – Fonctions et opérateurs usuels

A coté de cela, il ne faut pas oublier la commande `\sqrt`, qui peut prendre un argument optionnel pour l'ordre de la racine : $\sqrt{2}$

$$\sqrt[3]{x^2 + \exp y - 1} \quad \sqrt[3]{x^2 + \exp y - 1}$$

Enfin, nous allons voir les délimiteurs. Les délimiteurs sont des paires de symboles dont la taille s'adapte à ce qu'ils englobent. Cela concerne les parenthèses, les accolades, les crochets ; ...

On utilise pour cela la paire de commande `\left` et `\right`, chacune suivie du délimiteur souhaité qui peut être différent entre l'ouvrant et le fermant : `(,), [,], |, \{, \}`, ... On ne peut utiliser `\left` sans `\right` et inversement. Si l'on ne souhaite mettre un délimiteur que d'un coté, on utilisera le « . » de l'autre. Regardez la différence entre les 3 écritures d'une même formule dans l'exemple suivant :

Exemple 6.1 Utilisation des délimiteurs

```
\[
\{ (x^2+y^2)^2 \} = \left[ \{ (x^2+y^2)^2 \} \right] = \left[ \left[ \left[ (x^2+y^2)^2 \right] \right] \right]
\]
```

$$[(x^2 + y^2)^2] = \left[(x^2 + y^2)^2 \right] = \left[(x^2 + y^2)^2 \right]$$

6.3 Variantes de polices

On a vu qu'il existe 2 modes pour les formules mathématiques, et que le mode *displaymath* les écrivait de manière plus espacées et plus agréables. De même, les indices et les exposants sont écrits plus petits. On peut forcer la taille à celle de l'un de ces éléments.

`\displaystyle` permet dans un environnement *math* de forcer l'apparence de l'environnement *displaymath*

`\textstyle` permet dans un environnement *displaymath* de forcer l'apparence de l'environnement *math*

`\scriptstyle` permet de forcer la taille à celle d'un exposant

`\scriptscriptstyle` permet de forcer la taille à celle d'un exposant d'exposant

Par ailleurs, nous avons vu précédemment que l'on pouvait changer le style des caractères du texte. On peut aussi le faire dans les formules mathématiques :

`\mathrm` pour écrire en roman $\rightarrow a \quad \mathbf{a}$

`\mathit` pour écrire en italique $\rightarrow a \quad \mathbf{a}$

`\mathbf` pour écrire en gras $\rightarrow a \quad \mathbf{a}$

`\mathsf` pour écrire en sans sérif $\rightarrow a \quad \mathbf{a}$

`\mathtt` pour écrire en typewriter $\rightarrow a \quad \mathbf{a}$

`\mathbb` pour écrire en notation ensembliste $\rightarrow R \quad \mathbb{R}$

`\mathcal` pour écrire en calligraphique $\rightarrow C \quad \mathcal{C}$

Les symboles de fonctions (vus dans le Tableau 6.1) sont tous écrits en romain. Il en est de même par défaut sur tout ce qui se trouve en indice.

6.4 Symboles mathématiques

Je vais vous donner ici les commandes permettant d'utiliser les symboles mathématiques les plus courants. Pour une liste assez exhaustive, voir [29].

6.5 Lettres grecques

Pour les déclarations de variables, on utilise souvent les lettres grecques. La Tableau 6.2 liste les commandes de toutes les lettres grecques.

α	<code>\alpha</code>	ζ	<code>\zeta</code>	λ	<code>\lambda</code>	ρ	<code>\rho</code>	ϕ	<code>\phi</code>
β	<code>\beta</code>	η	<code>\eta</code>	μ	<code>\mu</code>	ϱ	<code>\varrho</code>	φ	<code>\varphi</code>
γ	<code>\gamma</code>	θ	<code>\theta</code>	ν	<code>\nu</code>	σ	<code>\sigma</code>	χ	<code>\chi</code>
δ	<code>\delta</code>	ϑ	<code>\vartheta</code>	ξ	<code>\xi</code>	ς	<code>\varsigma</code>	ψ	<code>\psi</code>
ϵ	<code>\epsilon</code>	ι	<code>\iota</code>	π	<code>\pi</code>	τ	<code>\tau</code>	ω	<code>\omega</code>
ε	<code>\varepsilon</code>	κ	<code>\kappa</code>	ϖ	<code>\varpi</code>	υ	<code>\upsilon</code>		

TABLE 6.2 – Les lettres grecques minuscules

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

TABLE 6.3 – Les lettres grecques majuscules

6.5.1 Symboles d'opérateurs et de relations

Certains symboles sont de taille variable suivant que l'on est en mode *math* ou *displaymath* (voir Tableau 6.5). C'est notamment le cas des symboles de somme (d'une série) et d'intégrale :

\pm	<code>\pm</code>	\times	<code>\times</code>	\div	<code>\div</code>	$*$	<code>\ast</code>
\star	<code>\star</code>	\circ	<code>\circ</code>	\bullet	<code>\bullet</code>	\cdot	<code>\cdot</code>
\cap	<code>\cap</code>	\cup	<code>\cup</code>	\oplus	<code>\oplus</code>		

TABLE 6.4 – Quelques symboles d'opérateurs

$\cap\cap$	<code>\bigcap</code>	$\sum\sum$	<code>\sum</code>	$\int\int$	<code>\int</code>
$\cup\cup$	<code>\bigcup</code>	$\prod\prod$	<code>\prod</code>	$\iint\iiint$	<code>\iint</code>
$\oplus\oplus$	<code>\bigoplus</code>	$\oint\oint$	<code>\oint</code>	$\iiint\iiint$	<code>\iiint</code>

TABLE 6.5 – Quelques symboles d'opérateurs de taille variable

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\subset	<code>\subset</code>	\supset	<code>\supset</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\neq	<code>\neq</code>	\perp	<code>\perp</code>
\propto	<code>\propto</code>	\sim	<code>\sim</code>	\simeq	<code>\simeq</code>	\approx	<code>\approx</code>

TABLE 6.6 – Quelques symboles de relations

Les symboles de relations peuvent être précédés de `\not` pour obtenir leurs opposés.

6.5.2 Symboles divers et flèches

Pour terminer ce tour rapide sur les symboles mathématiques, quelques symboles supplémentaires, et surtout les flèches.

\Re	<code>\Re</code>	\Im	<code>\Im</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>
∂	<code>\partial</code>	∞	<code>\infty</code>				

TABLE 6.7 – Quelques symboles divers

\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>		

TABLE 6.8 – Quelques flèches

6.5.3 Les accents

Il arrive fréquemment que l'on utilise des signes au dessus ou en dessous de la lettre définissant une variable, ce que l'on appelle des accents.

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\breve{a}	<code>\breve{a}</code>	\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>
\acute{a}	<code>\acute{a}</code>	\grave{a}	<code>\grave{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>

TABLE 6.9 – Les accents mathématiques

Toutefois, ces accents sont nécessairement de la largeur d'un caractère moyen. Pour définir des accents qui s'adaptent en largeur, voici la liste des plus courants :

\widehat{AB}	<code>\widehat{AB}</code>	\overline{AB}	<code>\overline{AB}</code>	\overrightarrow{AB}	<code>\overrightarrow{AB}</code>
\widetilde{AB}	<code>\widetilde{AB}</code>	\underline{AB}	<code>\underline{AB}</code>	\overleftarrow{AB}	<code>\overleftarrow{AB}</code>
\overbrace{AB}	<code>\overbrace{AB}</code>	\underbrace{AB}	<code>\underbrace{AB}</code>		

TABLE 6.10 – Les accents mathématiques de taille variable



La commande `\overbrace` accepte un commentaire en guise d'exposant. La commande `\underbrace` accepte un commentaire en guise d'indice.

Exemple 6.2 Ajouter des commentaires avec `\overbrace` ou `\underbrace`

```
\[
x^p = \overbrace{x \times x \times \cdots \times x \times x}^{p \text{ fois}}
      = \underbrace{x \times x \times \cdots \times x \times x}_{p \text{ fois}}
\]
```

$$x^p = \overbrace{x \times x \times \cdots \times x \times x}^{p \text{ fois}} = \underbrace{x \times x \times \cdots \times x \times x}_{p \text{ fois}}$$

6.6 Équations et systèmes d'équations

Jusqu'à présent, les environnements de mathématiques ne numérotent pas les équations. L'environnement qui le permet est *equation*.

Exemple 6.3 L'environnement *equation*

```
\begin{equation}
{(x+y)}^2 = x^2 + 2 x y + y^2
\end{equation}
```

$$(x + y)^2 = x^2 + 2xy + y^2 \quad (6.1)$$

Un numéro automatique est attribué et est affiché à droite.

 Il existe une version étoilée qui désactive la numérotation. Dans ce cas, l'environnement *equation** se comporte comme *displaymath*

On peut également souhaiter numérotter une suite d'équation qui font partie d'un même calcul en alignant par exemple certains éléments, comme le symbole =. On utilise pour cela l'environnement *align*

Exemple 6.4 L'environnement *align*

```
\begin{align}
{(2 x+y)}^2 &= {(2 x)}^2 + 2 (2 x) y + y^2 \\
&= 4 x^2 + 4 x y + y^2
\end{align}
```

$$(2x + y)^2 = (2x)^2 + 2(2x)y + y^2 \quad (6.2)$$

$$= 4x^2 + 4xy + y^2 \quad (6.3)$$

A l'intérieur de cet environnement, on insère des symboles & pour signifier un alignement et \\ pour passer à la ligne. L'alignement des colonnes est alterné entre un alignement à gauche pour les colonnes impaires et un alignement à droite pour les colonnes paires.

 On peut désactiver la numérotation pour certaines lignes en insérant la commande *\nonumber*, de préférence en fin de ligne pour la lisibilité.

Exemple 6.5 Utilisation de *\nonumber* dans un environnement *align*

```
\begin{align}
{(2 x+y)}^2 &= {(2 x)}^2 + 2 (2 x) y + y^2 \nonumber \\
&= 4 x^2 + 4 x y + y^2
\end{align}
```

$$(2x + y)^2 = (2x)^2 + 2(2x)y + y^2$$

$$= 4x^2 + 4xy + y^2 \quad (6.4)$$

 Il existe une version étoilée qui désactive complètement la numérotation.

Les environnements *equation* et *align* sont du même niveau que les environnements *math* et *displaymath*. Il existe toutefois un équivalent interne à l'environnement *align** : *aligned*. Ce qui signifie que l'on peut

le mettre autour d'une paire de délimiteurs par exemple, ou attribuer un numéro unique à l'ensemble, suivant dans quel environnement on l'utilise parmi *math*, *displaymath* ou *equation*.

Exemple 6.6 L'environnement *aligned*

```
\begin{equation}
\left\{
\begin{aligned}
(2x+y)^2 &= (2x)^2 + 2(2x)y + y^2 \\
(2x+y)^2 &= 4x^2 + 4xy + y^2
\end{aligned}
\right.
\end{equation}
```

$$\left\{ \begin{aligned} (2x+y)^2 &= (2x)^2 + 2(2x)y + y^2 \\ (2x+y)^2 &= 4x^2 + 4xy + y^2 \end{aligned} \right. \quad (6.5)$$

Il existe un autre environnement pour obtenir un résultat similaire, dans le cas uniquement d'un unique alignement par ligne (un seul & par ligne) : *cases*.

Exemple 6.7 L'environnement *cases* pour aligner des équations

```
\[
\begin{cases}
(2x+y)^2 &= (2x)^2 + 2(2x)y + y^2 \\
(2x+y)^2 &= 4x^2 + 4xy + y^2
\end{cases}
\]
```

$$\left\{ \begin{aligned} (2x+y)^2 &= (2x)^2 + 2(2x)y + y^2 \\ (2x+y)^2 &= 4x^2 + 4xy + y^2 \end{aligned} \right.$$

On constate une légère différence au niveau de l'espacement avant le symbole aligné : =. Cela nous ramène encore à la philosophie de \LaTeX , selon laquelle on distingue le rendu visuel de la structure du document. *cases* est dédié à écrire une alternative, avec éventuellement un texte qui suit que l'on souhaitera aligner, tandis que *aligned* est, comme *align* dédié à l'alignement d'équations. On pourra par exemple utiliser *aligned* pour définir un système d'équations dans lequel effectivement on alignera le symbole =.

Exemple 6.8 L'environnement *cases* pour écrire une alternative

```
\[
\begin{cases}
-\Delta u = f & \text{in } \Omega \\
u = 0 & \text{on } \Gamma = \partial\Omega
\end{cases}
\]
```

$$\left\{ \begin{aligned} -\Delta u &= f & \text{in } \Omega \\ u &= 0 & \text{on } \Gamma = \partial\Omega \end{aligned} \right.$$



Chaque ligne d'un environnement `cases` est en mode `math`. Pour éviter d'utiliser `\displaystyle` à chaque ligne, on peut utiliser l'environnement `dcases` qui le fait par défaut.

7.1 Inclusion d'images

Pour inclure une image dans un document L^AT_EX, on utilise la commande `\includegraphics` (fournie par l'extension `GRAPHICX`).

On précise la largeur et / ou la hauteur à l'aide des options respectives *width* et *height*

Exemple 7.1 Inclure une image

```
\includegraphics [width=2cm]{images/icons/focus.png}
\includegraphics [height=2em]{images/icons/focus.png}
\includegraphics [width=2cm, height=2em]{images/icons/focus.png}
```



Quand on compile avec `pdflatex`, les formats d'image autorisées sont `jpg`, `jpeg`, `png`, `pdf`. Par ailleurs, préciser l'extension n'est pas nécessaire en général.

7.2 Tableaux et matrices

Pour définir un tableau dans le corps de texte, on utilise l'environnement *tabular*

Exemple 7.2 Notre premier tableau

```
\begin{tabular}{ccc}
a & b & c \\
d & e & f \\
g & & \\
\end{tabular}
```

a	b	c
d	e	f
g		

Analysons l'exemple 7.2.

- Le premier argument de *tabular* définit à la fois le nombre de colonnes, mais aussi l'alignement du texte dans chaque cellule. Essentiellement 4 valeurs sont possibles :
 - l** pour une colonne de largeur automatique dans laquelle le texte sera aligné à gauche
 - r** pour une colonne de largeur automatique dans laquelle le texte sera aligné à droite
 - c** pour une colonne de largeur automatique dans laquelle le texte sera centré
 - p{}** pour une colonne de taille donnée entre accolades dans laquelle le texte sera justifié
- Chaque cellule d'une ligne est séparée par un `&`
- A la fin de chaque ligne de tableau, on utilise `\\`. On peut terminer une ligne prématurément, auquel cas les cellules restantes sont considérées vides.

On constate que le tableau résultant ne contient aucune bordure. Pour les rajouter, regardons l'exemple suivant :

Exemple 7.3 Tableaux et bordures pleines

```
\begin{tabular}{|p{1cm}|c|c|}
\hline
& aa & bb \\
\hline
\hline
xx & c & d \\
\hline
yy \\
\hline
zz & & \\
\hline
\end{tabular}
```

	aa	bb
xx	c	d
yy		
zz		

Que s'est-il passé ?

- On a inséré dans le premier argument de *tabular* des | pour chaque bordure verticale souhaitée.
- On a inséré `\hline` pour chaque bordure horizontale souhaitée.
- On peut doubler ces symboles pour avoir des bordures doubles
- Si une ligne du tableau se termine prématurément, les bordures horizontales ne sont pas affichées pour cette ligne. La seule solution est de définir les cellules correspondantes mais de n'y mettre aucun contenu.

Pour la première cellule du tableau de l'exemple 7.3, on souhaiterait que les bordures supérieures et gauche ne soient pas tracées. Pour cela, on peut procéder comme suit :

Exemple 7.4 Tableaux et bordures partielles

```
\begin{tabular}{|p{1cm}|c|c|}
\cline{2-3}
\multicolumn{1}{|l|}{} & \multicolumn{2}{c|}{aa} \\
\hline
xx & c & d \\
\hline
yy & e & f \\
\hline
\end{tabular}
```

	aa	
xx	c	d
yy	e	f

- La commande `\cline` permet de définir la longueur de la bordure horizontale à tracer en lui donnant la colonne de départ et la colonne d'arrivée.
- La commande `\multicolumn` permet de fusionner des cellules. Elle prend 3 arguments :
 1. La largeur de la cellule en nombre de colonnes ;
 2. L'alignement du texte dans la cellule ;
 3. Le contenu de la cellule.
- On peut en particulier utiliser `\multicolumn` pour effacer la bordure de gauche de la première cellule.

Nous avons parlé jusqu'ici de l'environnement *tabular* qui fonctionne dans le corps de texte, et pas en mode mathématique. Dans ce cas là, on utilise l'environnement *array*, de la même manière :

Exemple 7.5 Tableaux mathématiques

```

\begin{array}{|c|c|c|}
\cline{2-3}
\multicolumn{1}{|l|}{} & \multicolumn{2}{|c|}{aa} \\
\hline
xx & c & d \\
\hline
yy & e & f \\
\hline
\end{array}

```

	aa	
xx	c	d
yy	e	f

💡 Pour fusionner des cellules suivant les lignes, vous devez charger une extension supplémentaire, **MULTIROW** (voir [28]), qui fournit une commande `\multirow` qui a la même syntaxe que `\multicolumn`

7.3 Utiliser les environnements flottants

On peut souhaiter écrire une légende à une image ou un tableau. On utilise pour cela les environnements *figure* (pour les images) et *table* pour les tableaux.

Ces deux environnements fonctionnent de la même manière :

- Un numéro de figure ou de tableau est affecté.
- Ces 2 environnements sont dits « flottants » car leur positionnement dépend du corps de texte autour. On peut contrôler le positionnement avec un argument optionnel qui peut comprendre toute séquence composée d'un à 5 caractères parmi :
 - h** Si possible, la figure est insérée dans le corps de texte à l'endroit où l'environnement flottant est défini
 - t** Si possible, la figure est insérée en haut de la page contenant l'endroit où l'environnement est défini ou la page suivante
 - b** Si possible, la figure est insérée en bas de la page contenant l'endroit où l'environnement est défini ou la page suivante
 - p** La figure est insérée sur une page spéciale en contenant que des environnements flottants.
- ! Insister pour que \LaTeX respecte l'ordre indiqué par la séquence des autres caractères.

💡 Il se peut que \LaTeX ne puisse pas accéder à notre demande de positionnement, même avec le point d'exclamation. On peut contourner la difficulté en chargeant l'extension **FLOAT** qui nous donne une possibilité supplémentaire de placement : **H**, qui force à mettre l'environnement flottant là où on l'a défini, quitte à ajouter un saut de page, soit avant, soit après l'environnement flottant.

Exemple 7.6 Tableau flottant

```

\begin{table}[H]
\centering
\begin{tabular}{|c|c|c|}
\cline{2-3}
\multicolumn{1}{|1|}{} & \multicolumn{2}{|c|}{aa} \\
\hline
xx & c & d \\
\hline
yy & e & f \\
\hline
\end{tabular}
\caption{Exemple de tableau flottant}
\end{table}

```

	aa	
xx	c	d
yy	e	f

TABLE 7.1 – Exemple de tableau flottant

On peut lister les figures utilisées dans un document et les afficher, sur le même principe que la table des matières est une liste de titre et de sous-titres.

Pour afficher cette liste de figures, on utilise la commande `\listoffigures`.

De même on peut lister les tables à l'aide de la commande `\listoftables`.

C'est ainsi que sont construites ces listes en pages 3 et 4 de ce document.

7.4 Théorèmes

Quand on rédige un document scientifique, on peut certes vouloir écrire des formules mathématiques, mais on peut aussi vouloir définir des théorèmes. C'est là le rôle de l'extension `AMSTHM` dont je vais vous parler maintenant.

Pour définir un théorème, on utilise la commande `\newtheorem` :

```

\newtheorem{theorem}{Théorème}[chapter]

```

3 arguments obligatoires : le nom de l'environnement créé, son label et le niveau de numérotation. On dispose alors d'un environnement `theorem` (dans cet exemple), ayant un argument optionnel (pour préciser le nom du théorème ou son auteur par exemple), ce qui donne une fois utilisé un résultat de la forme :

Théorème 7.4.1 (Pythagore). *Dans un triangle rectangle, le carré de l'hypoténuse est égal à la somme des carrés des deux autres cotés.*

A.1 La liste complète des extensions utilisées

```

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[nomath]{kpfonts} % kpfonts autorise la combinaison \bfseries\scshape
\usepackage{geometry}
\usepackage{graphicx}
\usepackage{graphbox} % pour l'alignement vertical
\usepackage{xcolor}
\usepackage{calc}
\usepackage{mathtools}
\usepackage{latexsym}
\usepackage{amssymb}
\usepackage{amsthm}
\usepackage{fancyhdr}
\usepackage{fancyvrb}
\usepackage{enumitem}
\usepackage{tikz}
\usepackage{pgfplots}
\usepackage{xspace}
\usepackage{csquotes} % requis par biblatex
\usepackage[autocite=inline]{biblatex} % pour la bibliographie
\usepackage{float}
\usepackage{multirow}
\usepackage{tcolorbox}
\usepackage{etoolbox} % pour personnaliser les \tbclistof
\usepackage{ifthen}
\usepackage{lettrine}
\usepackage{babel}
\usepackage{microtype}
\usepackage{listings}
\usepackage[plainpages=false, pdfpagelabels, bookmarksnumbered, breaklinks=true,
  linkbordercolor=black, linkcolor=black, citecolor=black, colorlinks=true]{hyperref}
\usepackage{cleveref} % pour le referencement avec tcolorbox
\usepackage{minitoc}
\usepackage{todonotes}
\usetikzlibrary{fit, arrows, shadows, intersections, shapes, positioning, patterns, external}

```

Quelques mots sur les extensions utilisées :

INPUTENC concerne l'encodage des caractères. Cela doit se conformer à l'encodage du fichier \LaTeX que vous êtes en train d'écrire. Il est quand même fortement recommandé de prendre utf8 (même sous Windows pour lequel beaucoup d'éditeurs de textes restent en latin)

FONTENC concerne l'encodage de la police de caractères. *T1* est l'encodage le plus couramment utilisé. C'est pour dire que l'on utilise des polices de caractères de type 1.

KPFFONTS charge la police de caractères **KPFFONTS**. C'est l'une des polices recommandées avec **LMODERN**, et éventuellement **LIBERTINE** ou **FOURIER**. Je l'ai choisie car elle permet davantage de combinaisons que **LMODERN**, comme écrire du texte à la fois en petites capitales et en gras. Par contre, je n'aime pas la police utilisée pour les mathématiques. Fort heureusement, on peut la désactiver avec l'option *nomath*

GEOMETRY définit les marges d'une page, comme montré dans l'exemple en début de chapitre. Voir [7] pour plus de détails

GRAPHICX Cette extension concerne l'inclusion des images, et les formats d'images pris en comptes. Voir [4]

- GRAPHBOX** Cette extension augmente les possibilités d'inclusion d'images proposées par **GRAPHICX**, avec notamment l'alignement vertical. Voir [3]
- XCOLOR** Cette extension concerne la définition des couleurs, avec la possibilité de les mélanger de manière simple. Voir [18]
- CALC** Cette extension permet de réaliser des calculs arithmétiques. Voir [35]
- MATHTOOLS** Cette extension est dédiée à la mise en forme des environnements mathématiques. Elle corrige et complète l'extension **AMSMATH** qu'elle charge automatiquement. Voir [23]
- LATEXSYM** Cette extension basique charge les symboles mathématiques les plus courants.
- AMSSYMB** Cette extension charge également des symboles mathématiques et introduit notamment la commande `\mathbb` (vue en section 6.3).
- AMSTHM** C'est l'extension à charger pour définir des théorèmes. Voir [1]. Il existe une extension additionnelle **THMTOOLS** qui simplifie la personnalisation de l'apparence des théorèmes. Tout comme **TCOLORBOX** (voir plus bas dans la liste) ...
- FANCYHDR** Cette extension est dédiée à la personnalisation des en-têtes et pieds de page. Voir [27]
- FANCYVRB** Cette extension augmente les possibilités de manipulation des contenus *verbatim*. Je l'utilise conjointement avec **TCOLORBOX** pour pouvoir définir des propres environnements. Voir [40]
- FVEXTRA** Cette extension augmente les capacités de l'extension **FANCYVRB** en permettant des passages à la ligne automatiques pour les lignes très longues. Voir [30].
- ENUMITEM** Extension qui permet de définir complètement le style des listes (*itemize*, *enumerate* et *description*). Voir [21]
- TIKZ** Extension incontournable dédiée au graphisme et au dessin. Elle permet aussi de tracer des courbes à partir de fichiers de coordonnées, des arbres, des fractales et bien d'autres choses encore. TikZ est la surcouche utilisateur de l'extension **PGF**. Voir [13] pour accéder à leur documentation extrêmement complète et bien rédigée. Si je vous dis qu'elle fait plus de 1000 pages, vous aurez un aperçu de la puissance de cette extension.
- PGFPLOTS** Cette extension a pour but de faciliter davantage les tracés de fonctions par rapport à ce que propose déjà l'extension **TIKZ** sur laquelle elle s'appuie. Voir [12]
- XSPACE** Vous l'avez vu en section 4.1, l'appel d'une commande, comme `\TeX` et `\LaTeX`, n'est jamais suivi d'un espace par défaut; et si on veut en avoir un, il faut l'ajouter soit même. L'extension **XSPACE** introduit une commande intelligente `\xspace` qui décide en fonction de ce qui suit (mot, ponctuation, ...) s'il faut ajouter un espace ou non. Très utile lorsque l'on veut définir ses propres commandes ...
- CSQUOTES** Cette extension est dédiée à améliorer la gestion et la personnalisation des citations. Elle est notamment requise par l'extension **BIBLATEX**. Voir [38]
- BIBLATEX** C'est l'extension qui gère la bibliographie. Elle a été conçue pour rassembler et proposer davantage encore que les plupart des extensions qui existaient auparavant. Voir [19]
- FLOAT** Cette extension augmente les possibilités de positionnement des environnements flottants, notamment quand on veut absolument forcer un environnement flottant à être placé dans le corps de texte là où on l'a défini. Voir [20]
- MULTIROW** Cette extension ajoute la commande `\multirow` pour définir des cellules de tableaux sur plusieurs lignes. Voir [28]
- TCOLORBOX** Cette extension très puissante permet de définir des boîtes de couleurs autour de presque n'importe quoi. Elle permet aussi de redéfinir des équivalents de listings et de théorèmes. ... Voir [33]. Elle est utilisée abondamment dans ce document.
- ETOOLBOX** Cette extension permet des manipulations diverses de bas niveaux. Je l'ai utilisée dans ce document pour la mise en forme de ma liste d'exemples (extension **TCOLORBOX**) en souhaitant séparer par des lignes blanches les exemples pour matérialiser les chapitres. Voir [39]

- LETRINE** Cette extension est dédiée, comme son nom l'indique, à la manipulation de lettrines. Elle est utilisée ici pour positionner les images (loupe, ampoule, ...) dans les boîtes de dialogues. Voir [14]
- PDFPAGES** Cette extension introduit une commande `\includepdf` qui permet d'inclure une ou plusieurs pages d'un autre document pdf, comme si ces pages faisaient partie du document courant. Voir [30]
- MORESIZE** Cette extension introduit une commande de taille de caractères supplémentaire : `\HUGE`, utilisée dans cette présentation pour les watermarks. Voir [8]
- IFTHEN** Cette extension définit 2 commandes permettant de faire une boucle while et un test de type if-then-else. Voir [6]
- PGFOPTS** Cette extension est dédiée à la définition des arguments optionnels du chargement d'une extensions. Il y en a d'autres mais j'utilise celle-ci parce qu'elle est liée à l'extension **TIKZ**. Voir [37]
- BABEL** Cette extension charge les préférences de la langue dans laquelle vous allez écrire votre document. En effet, chaque langue a ses propres règles typographiques et cette extension est là pour les prendre en compte à la génération du document. Voir [22]
- MICROTYPE** Cette extension apporte des raffinements d'ordre typographique, comme l'ajustement de l'espacement permettant une meilleure coupure des mots. Le simple fait de charger cette extension suffit à avoir des améliorations notables. Voir [32] pour plus de détails.
- LISTINGS** Cette extension est à l'heure actuelle l'extension la plus complète pour insérer des listings de code, avec un nombre élevé de langages pré-définis, et la possibilité de définir ses propres langages pour la mise en page d'un listing. Voir [16]. Son seul véritable défaut est la non prise en compte des caractères accentués sauf dans des cas particuliers. Cette prise en compte est actuellement en cours de développement.
- HYPERREF** Cette extension est dédiée aux références croisées et à la possibilité de mettre des liens cliquables dans des documents. Voir [17].
- CLEVEREF** Cette extension complète et simplifie quelque peu l'utilisation de l'extension **HYPERREF**. Elle est utilisée ici pour pouvoir référencer mes boîtes de couleurs numérotées (extension **TCOLORBOX**). Voir [9]
- MINITOC** Cette extension est dédiée à la construction de tables des matières partielles attachées à une partie, un chapitre ou une section. Voir [11]
- TODONOTES** Cette extension permet de manipuler 2 commandes `\todo` et `\missingfigure` pour indiquer ce que l'on a prévu d'écrire à tel ou tel endroit dans le document en cours de rédaction. Le grand intérêt est de pouvoir afficher une liste de ces éléments avec la commande `\listoftodos` de sorte à savoir si on n'a effectivement rien oublié avant de finaliser notre document et le diffuser. Très pratique pour de gros documents comme ce support de cours. Voir [25]

- [1] *amsthm* – Typesetting theorems (AMS style). The American Mathematical Society. URL : <https://ctan.org/pkg/amsthm>.
- [2] Marc BAUDOIN. *Apprends L^AT_EX (5ème édition)*. URL : http://www.babafou.eu.org/Apprends_LaTeX/Apprends_LaTeX.pdf.
- [3] Niklas BEISERT. *graphbox* – Extend graphicx to improve placement of graphics. URL : <https://www.ctan.org/pkg/graphbox>.
- [4] David CARLISLE. *graphicx* – Enhanced support for graphics. URL : <https://ctan.org/pkg/graphicx>.
- [5] David CARLISLE, Morten HOGHOLM et The LaTeX TEAM. *xspace* – Define commands that appear not to eat spaces. URL : <https://ctan.org/pkg/xspace>.
- [6] David CARLISLE, Leslie LAMPORT et The LaTeX TEAM. *ifthen* – Conditional commands in LaTeX documents. URL : <https://www.ctan.org/pkg/ifthen>.
- [7] David CARLISLE et Hideo UMEKI. *geometry* – Flexible and complete interface to document dimensions. URL : <https://ctan.org/pkg/geometry>.
- [8] Christian CORNELSEN. *moresize* – Allows font sizes up to 35.83pt. URL : <https://www.ctan.org/pkg/moresize>.
- [9] Toby CUBITT. *cleveref* – Intelligent cross-referencing. URL : <https://www.ctan.org/pkg/cleveref>.
- [10] Xavier DANAUX. *moderncv* – A modern curriculum vitae class. URL : <https://www.ctan.org/pkg/moderncv>.
- [11] Jean-Pierre DRUCBERT. *minitoc* – Produce a table of contents for each chapter, part or section. URL : <https://ctan.org/pkg/minitoc>.
- [12] Christian FEUERSÄNGER. *pgfplots* – Create normal/logarithmic plots in two and three dimensions. URL : <https://www.ctan.org/pkg/pgfplots>.
- [13] Christian FEUERSÄNGER, Henri MENKE et Till TANTAU. *pgf* – Create PostScript and PDF graphics in T_EX. URL : <https://www.ctan.org/pkg/pgf>.
- [14] Daniel FLIPO. *lettrine* – Typeset dropped capitals. URL : <https://www.ctan.org/pkg/lettrine>.
- [15] Alexander GRAHN. *animate* – Create PDF and SVG animations from graphics files and inline graphics. URL : <https://www.ctan.org/pkg/animate>.
- [16] Jobst HOFFMANN, Carsten HEINZ et Brooks MOSES. *listings* – Typeset source code listings using L^AT_EX. URL : <https://ctan.org/pkg/listings>.
- [17] *hyperref* – Extensive support for hypertext in L^AT_EX. The LaTeX Team. URL : <https://ctan.org/pkg/hyperref>.
- [18] Uwe KERN. *xcolor* – Driver-independent color extensions for L^AT_EX and pdfL^AT_EX. URL : <https://ctan.org/pkg/xcolor>.
- [19] Philip KIME et Philipp LEHMAN. *BibLaTeX* – Sophisticated Bibliographies in L^AT_EX. URL : <https://ctan.org/pkg/biblatex>.
- [20] Anselm LINGNAU. *float* – Improved interface for floating objects. URL : <https://www.ctan.org/pkg/float>.
- [21] Javier Bezos LÓPEZ. *enumitem* – Control layout of itemize, enumerate, description. URL : <https://www.ctan.org/pkg/enumitem>.

- [22] Javier Bezos LÓPEZ et Johannes L. BRAAMS. *babel* – Multilingual support for Plain T_EX or L^AT_EX. URL : <https://www.ctan.org/pkg/babel>.
- [23] Lars MADSEN et al. *mathtools* – Mathematical tools to use with amsmath. URL : <https://ctan.org/pkg/mathtools>.
- [24] Andreas MATTHIAS. *pdfpages* – Include PDF documents in L^AT_EX. URL : <https://www.ctan.org/pkg/pdfpages>.
- [25] Henrik Skov MIDTIBY. *todonotes* – Marking things to do in a L^AT_EX document. URL : <https://www.ctan.org/pkg/todonotes>.
- [26] Frank MITTELBACH et Michel GOOSSENS. *The L^AT_EX Companion*. 2st. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2004. ISBN : 978-0-201-36299-2.
- [27] Piet van OOSTRUM. *fancyhdr* – Extensive control of page headers and footers in L^AT_EX2e. URL : <https://ctan.org/pkg/fancyhdr>.
- [28] Piet van OOSTRUM et Jerry LEICHTER. *multirow* – Create tabular cells spanning multiple rows. URL : <https://www.ctan.org/pkg/multirow>.
- [29] Scott PAKIN. *The Comprehensive L^AT_EX Symbol List* – Symbols accessible from L^AT_EX. URL : <https://www.ctan.org/pkg/comprehensive>.
- [30] Geoffrey POORE. *fvextra* – Extensions and patches for fancyvrb. URL : <https://www.ctan.org/pkg/fvextra>.
- [31] Axel REICHERT. *currvita* – Typeset a curriculum vitae. URL : <https://www.ctan.org/pkg/currvita>.
- [32] Robert SCHLICHT. *microtype* – Subliminal refinements towards typographical perfection. URL : <https://ctan.org/pkg/microtype>.
- [33] Thomas F. STURM. *tcolorbox* – Coloured boxes, for L^AT_EX examples and theorems, etc. URL : <https://ctan.org/pkg/tcolorbox>.
- [34] Till TANTAU et al. *beamer* – A L^AT_EX class for producing presentations and slides. URL : <https://ctan.org/pkg/beamer>.
- [35] Kresten Krab THORUP, Frank JENSEN et The LaTeX TEAM. *calc* – Simple arithmetic in L^AT_EX commands. URL : <https://www.ctan.org/pkg/calc>.
- [36] Didier VERNA. *CurVe* – A class for making curriculum vitae. URL : <https://www.ctan.org/pkg/curve>.
- [37] Joseph WRIGHT. *pgfopts* – LaTeX package options with pgfkeys. URL : <https://www.ctan.org/pkg/pgfopts>.
- [38] Joseph WRIGHT et Philipp LEHMAN. *csquotes* – Context sensitive quotation facilities. URL : <https://www.ctan.org/pkg/csquotes>.
- [39] Joseph WRIGHT et Philipp LEHMAN. *etoolbox* – e-_TE_X tools for L^AT_EX. URL : <https://www.ctan.org/pkg/etoolbox>.
- [40] Timothy Van ZANDT. *fancyvrb* – Sophisticated verbatim text. URL : <https://www.ctan.org/pkg/fancyvrb>.