

*pour bien débiter
quand on est
mathématicien·ne*

*TRÈS
COURTE
INITIATION À
L^AT_EX*

Ce document est mis à disposition selon les termes de la licence Creative Commons : « Attribution - Partage dans les mêmes conditions 4.0 International ».

Pour accéder à une copie de cette licence, merci de vous rendre à l'adresse suivante :

<https://creativecommons.org/licenses/by-sa/4.0/deed.fr>

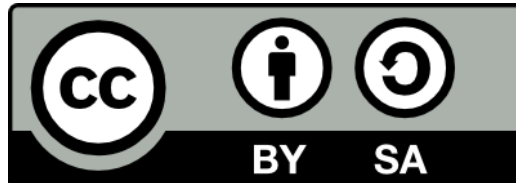


TABLE DES MATIÈRES

1	Introduction	7
1.1	Mais c'est quoi d'abord \LaTeX ?	8
1.2	De quoi allons nous parler?	8
1.3	WYSIWYG	8
1.4	Un peu de son histoire	9
1.5	\LaTeX et la typographie	9
1.6	Philosophie générale	10
1.7	Quelques avantages pour les documents mathématiques	10
1.7.1	Les mathématiques	10
1.7.2	Les très grands documents	11
1.7.3	Les bibliographies et les index	11
1.7.4	Un logiciel libre!	11
1.8	Des ressources et de l'aide	11
1.9	Les exemples	12
2	Installation	13
2.1	Tout un monde...	13
2.2	Les distributions	13
2.2.1	Avec Windows	13
2.2.2	Avec un Mac	14
2.2.3	Avec Linux, ou autres Unix	15
2.3	Un éditeur de texte	15
2.3.1	\TeX studio	15
2.3.2	D'autres éditeurs de texte qui peuvent servir à \LaTeX	16
3	Premiers pas	17
3.1	Édition du premier document	17
3.2	La compilation	18
3.2.1	<code>latexmk</code>	19
3.3	Dissection d'une commande \LaTeX	20

3.4	Explications du document	21
3.4.1	La classe du document	21
3.4.2	Le préambule	22
3.4.3	Le contenu du document	24
3.5	La structure des documents	24
3.5.1	Les mots	24
3.5.2	Les paragraphes	25
3.5.3	Le découpage du document	25
3.6	Les caractères spéciaux de \LaTeX	27
3.7	Changement de style	27
3.7.1	Emphase	29
3.8	Changement de corps (taille)	29
3.9	Des environnements structurants	30
3.9.1	Dissection d'un environnement	30
3.9.2	Les environnements quote et quotation	31
3.9.3	Les environnements center, flushright et flushleft	31
3.9.4	L'environnement verbatim	32
3.9.5	Les listes : itemize, enumerate et description	33
3.9.6	Les tableaux	35
3.10	Inclure des images	37
4	Flottants, références croisées, bibliographies et index	41
4.1	Références croisées	41
4.2	Titre de section	42
4.3	Flottants : table et figure	43
4.3.1	Listes des tableaux et des figures	44
4.4	Bibliographie – biblatex	45
4.4.1	Stocker sa base bibliographique	46
4.4.2	Charger le package et ajouter la bibliographie	49
4.4.3	Citer	49
4.4.4	Établir la bibliographie finale	51
4.4.5	Un ensemble d'outils	57
4.5	Index	59
4.5.1	Index simple	59
4.5.2	La commande <code>\index</code>	59
4.5.3	Formatage de l'index	60
5	Les mathématiques	63
5.1	Les modes mathématiques	63
5.1.1	Fonctionnement singulier	64
5.2	Les bases de la composition de mathématiques	65

5.2.1	Indices et exposants	65
5.2.2	Les espaces en mode mathématique	66
5.2.3	Du texte à l'intérieur d'une formule	66
5.2.4	Lettres grecques	67
5.2.5	Symboles d'opérateurs binaires	67
5.2.6	Des symboles, lettres et opérateurs utiles	68
5.2.7	Superposer deux symboles	72
5.2.8	Modification de style	72
5.2.9	Les fractions	74
5.2.10	Les racines	74
5.2.11	Délimiteurs	74
5.2.12	Symboles de taille variable	75
5.2.13	Fonctions et opérateurs	76
5.2.14	Les matrices	77
5.3	Les équations numérotées et les références	78
5.3.1	Les équations alignées	79
5.4	Théorèmes, définitions, etc.	80
6	Supports de présentation avec beamer	83
6.1	Préparation du document : le préambule	84
6.1.1	Les informations de son documents	84
6.2	Les diapositives	85
6.2.1	Diapositive de titre	86
6.3	Table des matières	87
6.4	Les blocs	88
6.5	Les couches	89
6.6	La bibliographie	91
7	Représentations graphiques	93
7.1	Utiliser le bon outil	93
7.1.1	\LaTeX pour des dessins « géométriques »	94
7.2	<code>Tikz</code>	94
7.2.1	<code>Tikz</code> pour l' impatient	94
7.2.2	Premier dessin	94
7.2.3	Le repérage des points	95
7.2.4	Dessiner et colorier	97
7.2.5	Les lignes	97
7.2.6	Le segment	97
7.2.7	Formes géométriques	100
7.2.8	Ajouter de annotations	100
7.3	<code>Pgfplots</code>	100

7.4	100
8 Packages utiles	101
8.1 La gestion des couleurs avec <code>xcolor</code>	101
8.1.1 Utiliser les couleurs	102
8.1.2 Charger des ensembles prédéfinis	102
8.2 Références croisées avec <code>refcheck</code>	103
8.3 Les unités du système international avec <code>siunitx</code>	104
8.4 Personnaliser les listes avec <code>enumitem</code>	105
8.4.1 Personnalisation de <code>itemize</code>	106
8.4.2 Personnalisation de <code>enumerate</code>	106
8.4.3 Personnalisation de <code>description</code>	107
8.5 Des beaux encadrements avec <code>tcolorbox</code>	107
8.6 Faciliter la composition mathématique avec <code>tdsfrmath</code>	109
8.7 Les dérivées avec <code>derivative</code>	110
8.8 Personnaliser les théorèmes <code>thmtools</code>	110
8.9 Composition avancée des tableaux	112
8.10 Utiliser le principe des TODO list avec <code>todonotes</code>	113
8.11 Faire des feuilles d'exercices avec <code>exesheet</code>	114
8.12 Écrire du code informatique avec <code>listings</code>	117
8.12.1 Utilisation standard	118
8.12.2 Personnalisation	119
8.13 Mettre en page des <i>pseudo codes</i>	121
8.13.1 Au début, <code>algorithms</code>	121
8.13.2 Le pseudo code	121
8.13.3 <code>algorithm2e</code>	123
9 Bibliographie	125
9.1 Les livres papiers ou PDF	125
9.2 Les liens internet	126
9.3 Les documentations	126
10 Les index	131

INTRODUCTION

Remarque 1.1 :

Attention, ce document est en cours de rédaction. Il est évidemment à améliorer, n'hésitez pas à me faire des retours par mail à l'adresse chupin@ceremade.dauphine.fr.

Je tiens à remercier Pierre CARDALIAGUET pour sa relecture.

Ce petit document se veut être une *très* courte introduction à \LaTeX . Vous verrez que les capacités de \LaTeX sont très importantes, et comme tout logiciel un peu sophistiqué, une si courte introduction ne suffit pas... Ce document existe donc pour permettre de mettre le pied à l'étrier avec le plus possible de *bonnes habitudes* et de *bonnes pratiques*. Il faudra compléter cette modeste ressource pour celles, très nombreuses, disponibles en livres et sur internet et qui seront indiqués comme références dans ce document dès que l'on sortira des fonctionnalités les plus classiques du logiciel.

Ce document s'adresse principalement aux étudiants et étudiantes en mathématiques découvrant le logiciel. Cependant, il pourra rester utile pour les enseignant-e-s, chercheurs et chercheuses du supérieur, avec notamment les chapitres des packages utiles (chapitre 8). De plus, pour ce type de population, l'apprentissage de ce logiciel se fait souvent *sur le tas*, et les mauvaises habitudes se prennent très rapidement.

Ainsi, ce document n'a pas vocation à couvrir tous les aspects de \LaTeX , mais

principalement de présenter ce qui est fondamental ou utile à la production de documents dans le domaine des mathématiques.

1.1 MAIS C'EST QUOI D'ABORD \LaTeX ?

Avant d'utiliser quelque chose, il est toujours très intéressant de s'intéresser un peu à l'histoire de la chose. Nous allons donc essayer de synthétiser quelques informations qui nous semblent importantes pour entrer dans le monde gigantesque de \LaTeX .

\LaTeX est un système de composition de document, c'est-à-dire qu'il permet de produire des documents, mais nous allons spécifier ça dans le reste du document.

1.2 DE QUOI ALLONS NOUS PARLER ?

Comme tout logiciel, il y a plusieurs niveaux d'approche. Nous ne rentrerons pas trop dans le détail ici, mais vous pouvez aller voir [2, 6] pour des introductions bien plus détaillées. Cette très légère introduction a largement (trop) été inspirée les diverses références, merci d'ailleurs à tous ces auteurs et autrices !

Définition 1.1 – \LaTeX

Lorsque nous parlerons de \LaTeX , nous entendrons à la fois un *langage* et un système de *composition de document*, c'est-à-dire un langage informatique et son *moteur* qui permet de produire des documents destinés à la lecture.

1.3 WYSIWYG

Le principe donc du système que nous présentons ici est :

- d'avoir un *fichier texte* dans lequel nous écrivons un ensemble de *commandes* et de texte qui décrivent le document à produire ;
- de *compiler* ce fichier texte pour produire le document à lire. C'est le programme qui permet de compiler que nous appelons *moteur*.

Le système de \LaTeX n'est donc pas un système WYSIWYG (*what you see is what you get*) comme l'est, par exemple, Libre Office Writer (ou Word).

1.4 UN PEU DE SON HISTOIRE

\LaTeX voit le jour en 1982 grâce à Leslie LAMPORT [24]. Il est en fait une *surcouche* de \TeX , qui est le programme noyau de \LaTeX , inventé par le mathématicien et informaticien Donald E. KNUTH en 1977 [22]. Pour la petite histoire, Donald KNUTH a écrit le logiciel \TeX pour la production de ses ouvrages *The Art of Computer Programming*. En effet, entre la parution du premier volume et du second, la composition au plomb a été abandonnée au profit de la photocomposition dont la qualité ne convenait pas à KNUTH (notamment pour la production de formules mathématiques).

Le nom \TeX vient du grec τέχνη, qui veut dire « art » et d'où vient le mot « technique ». Ceci explique la prononciation de \TeX et \LaTeX

Depuis son invention, \TeX et son monde ont continué d'évoluer et sont toujours très largement utilisés pour la production de documents scientifiques principalement. Il s'est largement imposé comme outil nécessaire dans le milieu des mathématiques, où toutes les publications scientifiques, ou presque, se font avec ce logiciel.

C'est parce qu'on est convaincu par la puissance de \LaTeX et parce que c'est la norme dans le monde des mathématiques que nous vous proposons cette très courte introduction.

1.5 \LaTeX ET LA TYPOGRAPHIE

Comme son histoire le montre, l'inventeur de \TeX est un féru de typographie. Très exigeant, son logiciel produit des documents d'une qualité typographique excellente.

Et avec \LaTeX , il n'y a pas besoin d'être un·e expert·e en typographie, nous verrons que beaucoup de chose sont gérées automatiquement par le logiciel. Parmi tout ça, la gestion des ligatures, des espaces différents, des séparations, des coupures de paragraphes, de pages, etc. Évidemment, il est aussi très intéressant de s'intéresser à ce merveilleux sujet qu'est la typographie, et pour cela, nous recommandons la lecture des *Petites leçons de typographie* de ANDRÉ [1].

Contrairement aux logiciels de traitement de texte qui fabriquent le document « au fur et à mesure », \LaTeX va traiter le document dans sa globalité à la compilation. Ceci permet d'être plus optimal et de produire des documents de meilleure qualité.

1.6 PHILOSOPHIE GÉNÉRALE

Même si tout cela est très abstrait pour le moment, il est important de comprendre la philosophie générale de la production de document par \LaTeX pour en faciliter son apprentissage.

Un document \LaTeX est un fichier texte simple, édité avec n'importe quel éditeur (\TeX studio, Notepad++, emacs, vim, etc.) et contenant le texte du document agrémenté de *commandes* \LaTeX qui le structurent.

Lors de la frappe, on ne voit pas le document final produit au fur et à mesure, il faut compiler le document pour obtenir le fichier PDF qu'on peut visualiser. C'est un paradigme difficile à adopter lorsqu'on a toujours été habitué aux logiciels WYSIWYG, mais qui peut se révéler productif. En effet, cela permet de se concentrer uniquement sur le fond, le contenu du comment, sa structure sans se préoccuper des soucis de mise en pages. Mais il faut savoir lâcher prise et laisser faire le logiciel.¹

Pour produire un document, le *moteur* traite le document texte, produit des petites boîtes (par exemple pour un caractère) et des plus grosses (par exemple, un paragraphe), qu'il emboîte, juxtapose, assemble, en mettant des ressorts entre chaque qui peuvent se compresser ou se dilater. Une fois une page remplie, le moteur passe à la composition de la page suivante. Cela ressemble à la production des pages au plomb. Comprendre cela, permet de mieux comprendre les erreurs lorsqu'il y en a.

1.7 QUELQUES AVANTAGES POUR LES DOCUMENTS MATHÉMATIQUES

En plus d'être la norme dans le milieu des mathématiques, \LaTeX a des avantages importants sur d'autres solutions de composition.

1.7.1 Les mathématiques

Inventé par un mathématicien, il est parfaitement adapté pour l'écriture de formule mathématiques complexes avec là encore une qualité typographique impeccable. Le chapitre 5 est dédié à ce sujet.

1. Évidemment, avec du temps, des efforts, et une bonne connaissance de \LaTeX , il est possible de tout modifier de l'apparence de document !

1.7.2 Les très grands documents

Il est parfaitement adapté à la production d'énorme document (thèse, mémoire, etc.) avec des mécanismes de références puissants (que nous présenterons dans le chapitre 4). Le logiciel ne « plante » pas.

1.7.3 Les bibliographies et les index

Nous présenterons en section 4.4 du chapitre 4 aussi le fonctionnement d'outils qui permettent la gestion des bibliographies et des index. Là encore, \LaTeX permet d'automatiser beaucoup de choses.

1.7.4 Un logiciel libre!

C'est un logiciel libre! Avec une énorme communauté d'utilisateurs et utilisatrices qui développent et proposent des solutions à quasi tous les problèmes qu'on peut se poser.

De plus, le développement de \LaTeX assure la rétro-compatibilité, ce qui permet de compiler des documents qui ont plus de 20 ans!

1.8 DES RESSOURCES ET DE L'AIDE

Les ressources, les documentations, et l'aide sont nombreuses. Une large communauté d'utilisateurs et d'utilisatrices permettent d'avoir des ressources et de l'aide pour les problèmes que l'on rencontre.

Plusieurs sites sont des ressources importantes. Outre le très connu, mais anglophone, <https://tex.stackexchange.com/> [42], nous avons l'équivalent francophone [36] avec

<https://texnique.fr/osqa/>

qui est un site collaboratif, où on peut poser des questions sur \LaTeX . N'hésitez pas, une grande communauté d'utilisateurs et d'utilisatrices se feront un plaisir de vous aider.

Par ailleurs, Denis BITOUZÉ a produit un grand nombre de documents pédagogique sur l'utilisation de \LaTeX [5]

<http://gte.univ-littoral.fr/members/dbitouze/pub>

Une association des utilisateurs francophones de \TeX , GUTenberg, existe aussi [23], avec là encore, un site internet qui regroupe des sources d'information sur le logiciel :

<https://www.gutenberg.eu.org/>

1.9 LES EXEMPLES

Attention, si ce document veut illustrer le plus possible l'utilisation de \LaTeX au travers d'exemples, les résultats présentés ne ressembleront pas tout à fait à ce que vous obtiendrez par défaut avec \LaTeX . En effet, pour la composition de cet ouvrage, de nombreux paramètres ont été modifiés comme la police d'écriture, l'apparence des titres, les couleurs, etc.

INSTALLATION

2.1 TOUT UN MONDE...

\LaTeX fait partie d'un vaste monde. Comme expliqué en introduction, il y a plusieurs aspects au *système* de production : un langage, un moteur, une compilation, etc. C'est pour cela que pour pouvoir utiliser \LaTeX , il est nécessaire d'installer une *distribution* qui est un ensemble de programmes.

2.2 LES DISTRIBUTIONS

La très rapide description qui suit est largement tirée de la présentation de Denis BITOUZÉ [4] disponible ici :

<https://gte.univ-littoral.fr/Members/denis-bitouze/pub/latex/diapositives-cours-d/conference-n-1/@@download/file/en-ligne1.pdf>

Si les instructions qui suivent ne vous suffisent pas, allez regarder la partie installation de ce fichier PDF.

2.2.1 Avec Windows

Sous Windows, nous conseillons l'installation de la distribution \TeX Live :

<https://www.tug.org/texlive/>

Installation avec téléchargement au fur et à mesure

Si votre connexion internet est fiable, vous pouvez suivre la procédure décrite ici. Sinon, il est préférable de faire l'installation à partir de l'image disque comme décrit dans la section suivante.

Il faudra dans un premier temps, télécharger le fichier `install-tl-windows.exe` disponible à cette page :

<https://www.tug.org/texlive/acquire-netinstall.html>

Exécuter ce fichier sur votre système. Une boîte de dialogue peut apparaître pour conseiller de désactiver l'anti-virus le temps de l'installation. Nous conseillons, en suivant les conseils de Denis BITOUZÉ [4], de :

1. tenter l'installation avec l'anti-virus activé, en cliquant sur « Continuer » ;
2. si l'installation échoue, retenter l'installation avec l'anti-virus désactivé.

Installation à partir de l'image disque (ISO)

Vous pouvez aussi installer la distribution à partir d'un fichier image (`texlive2023xxx.iso`¹ téléchargeable ici :

<https://ctan.gutenberg.eu.org/systems/texlive/Images/>

Il vous faudra cependant installer un logiciel (libre) de montage d'image tel que WinCDEmu

<https://wincdemu.sysprogs.org/>

Il faudra ensuite *monter* l'image avec un clic droit sur le fichier `texlive2020xxx.iso` en sélectionnant *Monter*. Ainsi vous aurez accès à l'installateur `install-tl-windows`.

Vidéo de démonstration de Denis BITOUZÉ

La procédure est illustrée dans la vidéo suivante :

<https://www.youtube.com/watch?v=SXoErhQlkDg&feature=youtu.be>

2.2.2 Avec un Mac

Sous MacOS, nous conseillons l'installation de la distribution MacTeX

<https://www.tug.org/mactex/>

Ici l'installation est plus simple puisqu'il suffit de télécharger le fichier `MacTeX.pkg` puis de double cliquer dessus et de suivre les instructions.

1. Les xxx dépendent de la dernière mise à jour et l'année peut elle aussi changer.

2.2.3 Avec Linux, ou autres Unix

Si vous êtes sous Linux, pour installer la distribution $\text{T}_{\text{E}}\text{X}$ Live, il suffira de télécharger l'archive `install-tl-unx.tar.gz`, d'extraire l'archive puis, dans un terminal, de lancer, avec les droits *super utilisateur* (sudo avec Ubuntu) :

```
user #> ./install-tl
```

et de suivre les instructions. Il est très commode de créer les liens symboliques vers les répertoires standard `/usr/local/bin`, `/usr/local/man`, et `/usr/local/info`.

Là encore, la page

<https://www.tug.org/texlive/quickinstall.html>

décrit, en anglais, la procédure. Vous pouvez aussi vous rapporter au PDF [4] de Denis BITOUZÉ pour une description en français.

2.3 UN ÉDITEUR DE TEXTE

Il existe une multitude d'éditeurs de texte plus ou moins adaptés à l'édition de sources pour $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Chacun·e doit trouver chaussure à son pied.

Pour les moins aguerris·es, nous en proposons quelques uns.

2.3.1 $\text{T}_{\text{E}}\text{X}$ studio

$\text{T}_{\text{E}}\text{X}$ studio est un environnement de développement intégré pour $\text{T}_{\text{E}}\text{X}$. À cet égard, il n'est conçu que pour écrire des fichiers pour le monde de $\text{T}_{\text{E}}\text{X}$. Mais cela a beaucoup d'avantages, notamment pour les débutant·e·s. En effet, il permet :

- de visionner le résultat final facilement ;
- de déboguer facilement en cas d'erreur à la compilation ;
- de compiler facilement un document ;
- de colorier syntaxiquement le code ;
- de compléter automatiquement les commandes usuelles de $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ce qui est accélère considérablement l'édition ;
- et d'autres choses encore.

De plus, il s'installe sur tous les systèmes d'exploitations. Nous vous recommandons donc fortement son installation :

<https://www.texstudio.org/>

2.3.2 D'autres éditeurs de texte qui peuvent servir à \LaTeX

Parmi tous les éditeurs de texte qui peuvent servir pour l'édition de document avec \LaTeX , en voici une liste non exhaustive regroupant des éditeurs de qualité :

Éditeur	OS	Type
Emacs	muli OS	généraliste
Gedit	Linux	généraliste
Kate	muli OS	généraliste
Nano	Linux	généraliste
Notepad(++)	Windows	généraliste
Overleaf	service web	\LaTeX
Sublime Text	multi OS	généraliste
\TeX maker	multi OS	\LaTeX
\TeX nic Center	Windows	\LaTeX
\TeX shop	MacOS	\LaTeX
Vim	multi OS	généraliste
VScode	multi OS	généraliste

PREMIERS PAS

Après cette présentation très générale, on aborde les premiers pas avec le système de composition. On développera les points spécifiques dans les chapitres suivants.

3.1 ÉDITION DU PREMIER DOCUMENT

On présente ici un document *minimal* qui permet la production d'un document. Il est tout à fait normal que ce document ne veuille rien dire pour les néophytes. Pour arriver à comprendre ce code, nous allons le décortiquer.

```
\documentclass[french,a4paper]{article} % la classe du document
\usepackage[utf8]{inputenc} % encodage des caractères
\usepackage[T1]{fontenc} % encodage de la fonte
\usepackage{geometry} % la gestion de la géométrie de la page
\usepackage{mathtools,amssymb} % pour tous les maths
\usepackage{graphicx} % pour la gestion des images
\usepackage{babel} % gestion des langues
\usepackage{ntheorem,thmtools} % pour les théorèmes
\usepackage{hyperref} % les liens hypertextes
```

```
\begin{document}

Voici mon premier document \LaTeX !

\end{document}
```

Remarque 3.1 :

Comme tout langage de programmation, \LaTeX a, lui aussi, un système de commentaire. Les commentaires ne seront pas imprimés. Ils peuvent se révéler très utiles, notamment pour se donner des explications ou des points de repère dans notre document.

Un commentaire commence par un % et se termine à la fin de la ligne :

```
Ceci est du texte. % ceci est un commentaire
La suite du texte.
```

Supposons, à partir de maintenant, que l'on a sauvegardé ce document sous le nom `premier.tex` (les documents \LaTeX ont l'extension `.tex`).

3.2 LA COMPILATION

Si vous utilisez \TeX studio, cette partie se résume à cliquer sur le bouton de « production et visualisation » (bouton avance-rapide) comme le montre la figure 3.1.

Cependant, il est intéressant de connaître et comprendre ce qu'il se passe derrière ce bouton.

Dans un terminal, l'obtention du document PDF à partir du source `.tex` s'obtient grâce à la commande suivante :

```
user $> pdflatex premier.tex
```

Une telle compilation produit plusieurs fichiers. Tout d'abord le document `premier.pdf` qui est le document que l'on souhaite produire qui nous montrons en figure 3.2.

D'autres fichiers sont produits lors de la compilation comme `premier.log`, `premier.aux`, `premier.out`, `premier.thm`, etc. Pour ne pas trop rentrer dans les

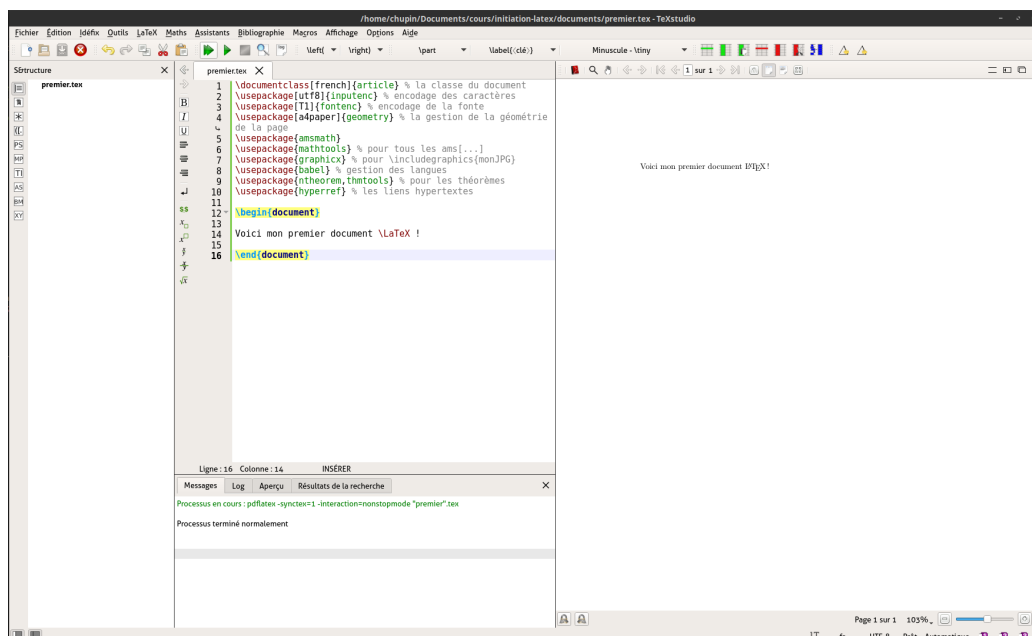


FIGURE 3.1 – Le fichier `premier.tex` dans \TeX studio.

détails, le fichier de *log* est crucial. Comme tout fichier de *log* (journal), il contient tous les événements affectant le processus de compilation. C’est en quelque sorte la sortie de compilation, et donc potentiellement, c’est dans ce fichier-là que seront notées les erreurs de compilation. Nous vous invitons donc à aller y jeter un œil.

On peut noter l’existence de \Lua\TeX qui peut remplacer \pdf\TeX . Ce moteur est le successeur de \pdf\TeX et peut être utilisé très simple. De plus, il permet l’utilisation native du codage UTF-8, ainsi que l’utilisation de fontes modernes telles que celles en OpenType.

3.2.1 latexmk

`latexmk` est un programme qui permet, en ligne de commande, d’automatiser tout un tas d’opérations nécessaires à l’obtention du document final. On verra, par exemple dans le chapitre 4, qu’il est quelquefois nécessaire de compiler plus fois, voir de faire appel à un programme externe durant la chaîne de compilation (pour les bibliographies, les index, etc.). Le programme `latexmk` se charge de tout cela pour vous, et est hautement paramétrable pour faire un grand nombre d’opérations. Sa présentation dépasse le cadre de cette introduction. Les éditeurs dédiés à \LaTeX ont, eux aussi, leurs mécanismes permettant d’automatiser les compilations pour obtenir un document final.

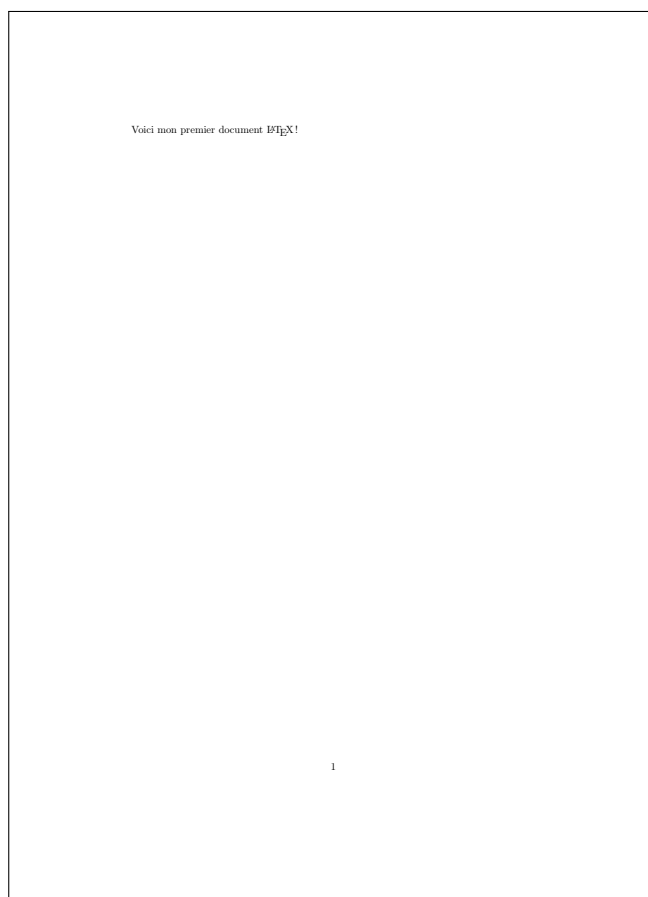


FIGURE 3.2 – Fichier PDF produit par la compilation `pdflatex`

3.3 DISSECTION D'UNE COMMANDE LATEX

Pour pouvoir décortiquer ce mini document, il nous faut expliquer la syntaxe générale d'une commande de base L^AT_EX. L^AT_EX étant un langage informatique, il est composé de commandes.

Définition 3.1 – Commande

Voici la structure d'une commande \LaTeX .

\backslash NomDeCommande[\langle options \rangle]{ \langle argument \rangle }

- Une commande commence par un *backslash* appelé caractère d'échappement.
- Ce *backslash* est suivi par le nom de la commande.
- Suivant la définition des commandes, certaines acceptent, entre crochets, des arguments optionnels.
- Les arguments des commandes se trouvent entre accolades.

Par exemple, on aura des commandes sans argument comme \centering , des commandes avec argument comme $\text{\section\{Introduction\}}$, et des commandes avec argument optionnel comme $\text{\includegraphics[width=10cm]\{monimage.pdf\}}$.

Nous reviendrons au fur et à mesure de ce document sur d'autres subtilités des commandes en \LaTeX .

3.4 EXPLICATIONS DU DOCUMENT

3.4.1 La classe du document

Le fichier `premier.tex` commence par la commande

$\text{\documentclass}\langle$ arguments \rangle {`article`}

Il est très important de retenir que tout document \LaTeX commence par cette commande.

L'argument obligatoire

Son argument obligatoire (ici `article`), définit la classe du document. Voici la liste des principales classes :

article : est la classe plutôt dédiée aux documents courts (rapports, TP, feuille d'exercices, etc.);

report : est la classe utilisée pour les documents plus longs qui peuvent être découpé en *chapitre*, la classe `article` ne gère pas les chapitres.

book : est la classe pour la rédaction de livres. Il y a en fait que peu de différence avec la classe `report`.

Les options de la classe

Il existe des options de classe (qui vont dépendre de la classe choisie).

Parmi les plus importantes :

11pt : qui porte la taille des caractères à onze points (unité de mesure utilisée en typographie). Il existe aussi l'option **12pt**. La taille par défaut est à **10pt** (avec l'option correspondante).

twocolumn : qui permet de composer le document sur deux colonnes.

twoside : qui permet de régler les marges du document pour une impression recto-verso (dans quel cas, la marge intérieure sera plus petite que la marge extérieure).

Certaines options sont données à la classe mais sont en réalité transférée aux packages chargés ensuite. C'est le cas des options présentées dans le code introductif de ce chapitre, par exemple :

french : qui, via le package `babel`, charge les normes typographiques françaises¹ ;

a4paper : qui, via le package `geometry`, ajuste la disposition de la page pour le format A4².

3.4.2 Le préambule

Le préambule d'un document \LaTeX est tout ce qui se trouve entre le `\documentclass` et le `\begin{document}`. Dans le cas simple du document `premier.tex`, il sert à charger des *packages* (ou extensions en français).

Sans rentrer dans le détail, faisons un rapide tour de l'ensemble assez restreint que nous proposons dans le fichier `premier.tex`.

Quelques extensions (ou packages) de bases

\LaTeX fournit tout un ensemble de commandes qui permettent de faire déjà beaucoup de chose. Cependant, depuis sa création, il a fallu répondre à de nouveaux besoins (langues, format de page, etc.), aux nouvelles technologies, et à bien d'autres choses encore.

Une extension (ou *package* en anglais) permet d'ajouter des fonctionnalités à \LaTeX soit en redéfinissant le fonctionnement de base, soit en ajoutant des commandes.³

1. Cette option permet aussi de spécifier à d'autres packages qui agissent différemment suivant la langue utilisée, que le document est en français.

2. Aux États-Unis, ce n'est pas le format usuel, d'où, la nécessité d'ajuster le comportement par défaut de \LaTeX .

3. Ce mécanisme est très riche et permet aux utilisateurs et utilisatrices de \LaTeX d'enrichir le logiciel et de le proposer aux autres.

Ces extensions sont *chargées* grâce à la commande `\usepackage`. Faisons un tour rapide des packages utilisés dans le document `premier.tex` et qui pourront vous être utiles.

inputenc : Cette extension [19] permet de spécifier le *codage* du document texte. Avec tous les systèmes d'exploitations récents, l'UTF-8 est la norme et cela se spécifie avec l'option `utf8`. L'encodage du document est réglé par votre éditeur de texte. Dans d'anciens documents, on peut encore rencontrer l'option `latin9` pour l'encodage ISO 8859-1⁴.

fontenc : Cette extension [45] avec l'option `T1` spécifie le codage des caractères du document produit (PDF). Cela permet une meilleure gestion des caractères accentués par exemple.

geometry : Cette extension [48] permet de gérer les différentes dimensions des pages. L'option `a4paper` donné à la classe, mais transmise au package `geometry`, permet de fournir des réglages pour le papier a4 (qui n'est pas la norme aux États-Unis par exemple).

mathtools, amssymb : Ces extensions [`ctan-amssymb`, 16] ajoutent les commandes et symboles mathématiques pour à peu près tout écrire. Le package `mathtools` charge et corrige certains bugs de `amsmath` en apportant au passage quelques fonctionnalités supplémentaires. Les packages `amsmath` et `amssymb` ont été produits par l'*American Mathematical Society* (AMS).

graphicx : Cette extension [9] permet en particulier la gestion des images, et des inclusions d'images (jpg, png, pdf, eps, etc.).

babel : Cette extension [27] permet de gérer les langues et les conventions typographiques allant avec. Avec l'option `french`, chargée dans la classe de document mais transmise à `babel`, on obtient les bons espaces, les bonnes coupures de mots, les bons titres, etc. le tout, automatiquement (voir [13]).

ntheorem, thmtools : Ces extensions [30, 41] permettent de composer facilement des théorèmes, des définitions, des propositions, et d'en personnaliser l'apparence.

hyperref : Cette extension [37] permet de profiter des possibilités de navigation au sein d'un document PDF comme les références croisées, les références bibliographiques, les notes, les URL, etc. Elle permet aussi d'avoir une table des matières navigable.

Nous conseillons de charger ces extensions dans tous vos documents. Notons aussi que l'ordre de chargement des extensions a son importance, et que l'ordre proposé ici est fonctionnel. Enfin, il existe des milliers d'extensions qui permettent

4. Depuis 2018, l'encodage par défaut avec \LaTeX est l'UTF-8, il n'est donc plus nécessaire de charger `inputenc` avec l'option `utf8`

de faire toutes sortes de choses. Le catalogue des extensions est le **CTAN** et se trouve à l'adresse suivante

<https://www.ctan.org/>

Les informations disponibles sur internet sont immenses, et même s'il est parfois difficile de s'y retrouver, il est nécessaire d'aller y chercher les réponses aux problèmes que l'on se pose. Nous essayons dans ce document de répondre aux problèmes et questionnements les plus courants et les plus basiques pour la composition de documents en lien avec les mathématiques⁵. Une fois le pied mis à l'étrier, les réponses sur le web se trouvent sans difficulté.

3.4.3 Le contenu du document

Le contenu qui sera composé par \LaTeX se trouve lui entre le `\begin{document}` et `\end{document}`. Tout texte placé après `\end{document}` est simplement ignoré.

Dans le document `premier.tex`, le contenu est court et simple :

```
Voici mon premier document \LaTeX{}
```

3.5 LA STRUCTURE DES DOCUMENTS

3.5.1 Les mots

La brique de base de la composition de document \LaTeX est le mot. Les mots sont séparés par des signes de ponctuation ou des espaces. Il est important de noter qu'un simple retour à la ligne est considéré comme un espace. Un ou plusieurs espaces produisent le même résultats, vous pouvez ainsi mettre autant d'espaces que vous le désirez comme le montre les exemples suivants.

```
Voici un texte qui est écrit simplement  
avec  
un retour tout de même.
```

```
Voici un texte qui est écrit simplement avec un retour tout de même.
```

5. Notons cependant que les explications, les conseils et méthodes proposés ici restent valables pour d'autres domaines.

```
En voici un
    autre un peu
plus alambiqué !
```

```
-----
En voici un autre un peu plus alambiqué!
```

3.5.2 Les paragraphes

Un document \LaTeX est composé de paragraphes qui sont des ensembles de mots séparés les uns des autres par au moins une ligne *vide*. Là encore, une ou plusieurs lignes vides produiront le même résultat.

```
Premier paragraphe.
```

```
Ce deuxième paragraphe va permettre encore
une fois de montrer que
les retours de ligne simples n'ont aucune importance sur le
    formatage
du paragraphe.
```

```
Une ou trois lignes séparent de la même manière les paragraphes.
```

```
Premier paragraphe.
```

```
Ce deuxième paragraphe va permettre encore une fois de montrer que les retours
de ligne simples n'ont aucune importance sur le formatage du paragraphe.
Une ou trois lignes séparent de la même manière les paragraphes.
```

Comme on le constate sur l'exemple, malgré les lignes vides dans le code source, les paragraphes eux ne sont pas séparés par une ligne vide. Il existe quelques commandes qui ajoutent des espaces verticaux entre paragraphes, mais elles sont à éviter. En effet, la philosophie de \LaTeX est de ne pas s'intéresser à la forme mais juste au contenu. Pour information, ces commandes sont `\smallskip`, `\medskip`, `\bigskip` par ordre croissant de taille.

3.5.3 Le découpage du document

Dans toutes les classes standards, un document se découpe en *section*, *sous-section*, *sous-sous-section*, etc. le tout se numérotant automatiquement.

Les commandes sont

```
\section{<titre de la section>}
\subsection{<titre de la sous-section>}
\subsubsection{<titre de la sous-section>}
```

Et viennent ensuite les paragraphes qui ne sont pas numérotés.

```
\paragraph{<titre du paragraphe>}
\subparagraph{<titre du sous paragraphe>}
```

Les classes book et report permettent de découper le document à l'aide de parties via la commande :

```
\part{<titre de la partie>}
```

et de découper les parties (si besoin) en chapitres via la commande :

```
\chapter{<titre du chapitre>}
```

Remarque 3.2 :

Il existe des versions « étoilées » de ces commandes, qui ne génèrent pas de numéro de partie/chapitre/section : `\part*`, `\chapter*`, etc.

La table des matières

Avec le découpage du document par ces commandes, on peut très facilement générer la table des matières.

Cela se fait avec la commande :

```
\tableofcontents
```

qui suivant la classe du document va générer une section ou un chapitre dédiée à la table des matières. Attention cependant, le chapitre ou la section sont alors *non numérotés*.

Remarque 3.3 :

Attention, les parties, chapitres et autres sections produits par des commandes *étoilées* ne figurent pas dans la tables de matières.

3.6 LES CARACTÈRES SPÉCIAUX DE \LaTeX

Dans le code source, certains caractères ont un comportement spécial comme, par exemple, le caractère % qui permet de commenter le code. De même, le caractère \ produit les commandes. En fait, il existe dix caractères au comportement spécial :

```
$ & % # _ { } ~ ^ \
```

Ces caractères ne peuvent pas être imprimés tels quels. Pour sept d'entre eux, on peut les imprimer dans le texte grâce au caractère \ lui-même :

```
\$ \& \% \# \_ \{ \} \~ \^ \
```

```
$ & % # _ { } #
```

Pour les trois autres, on peut utiliser les commandes :

```
\textasciitilde \textasciicircum  
\textbackslash
```

```
~^\

---


```

3.7 CHANGEMENT DE STYLE

Il est souvent souhaitable de varier le style de composition (gras, italique, incliné, etc.) pour hiérarchiser, mettre en emphase, etc. Pour cela, il existe des commandes que nous synthétisons dans le tableau 3.1⁶.

Notons que ces commandes modifient uniquement le style de leur argument, le texte qui suit n'est pas affecté.

```
L'italique sert souvent à mettre un mot en \textit{emphase}.  
Mais  
comment mettre un mot en emphase dans un texte en italique ?
```

L'italique sert souvent à mettre un mot en *emphase*. Mais comment mettre un mot en emphase dans un texte en italique ?

6. Pour la composition de ce document la fonte utilisée, Libertinus, n'est pas celle de base de \LaTeX (qui est Latin Modern). Pour l'exemple, le tableau 3.1 est composé avec Latin Modern.

Commande	Résultat	Signification
<code>\textup{droit}</code>	droit	upright
<code>\textit{italique}</code>	<i>italique</i>	<i>italic</i>
<code>\textsl{incliné}</code>	<i>incliné</i>	<i>slanted</i>
<code>\textsc{petites capitales}</code>	PETITES CAPITALES	SMALL CAPS
<code>\textmd{maigre}</code>	maigre	medium
<code>\textbf{gras}</code>	gras	boldface
<code>\textrm{romain}</code>	romain	roman
<code>\textsf{linéal}</code>	linéal	sans serif
<code>\texttt{machine à écrire}</code>	machine à écrire	typewriter

TABLE 3.1 – Commandes de changement de style.

Bien entendu, on peut cumuler les modification de style, on peut par exemple produire de l’italique gras.

```
\textit{Qu’est-ce donc que \textbf{cela} ?}
```

*Qu’est-ce donc que **cela** ?*

Il existe aussi des *déclarations* permettant d’effectuer les même changements de style. Les déclarations ne prennent pas d’argument et agissent sur l’ensemble du texte qui suit la déclaration. Pour en limiter la portée, il faut utiliser *des accolades de groupement* comme le montre l’exemple suivant.

Un paragraphe dans le style par défaut.

```
{\itshape Un paragraphe composé en italique.}
```

Les accolades de groupement faisant leur office, ce troisième paragraphe est dans le style par défaut.

Un paragraphe dans le style par défaut.

Un paragraphe composé en italique.

Les accolades de groupement faisant leur office, ce troisième paragraphe est dans le style par défaut.

Commande	Déclaration
<code>\textup</code>	<code>\upshape</code>
<code>\textit</code>	<code>\itshape</code>
<code>\textsl</code>	<code>\slshape</code>
<code>\textsc</code>	<code>\scshape</code>
<code>\textmd</code>	<code>\mdseries</code>
<code>\textbf</code>	<code>\bfseries</code>
<code>\textrm</code>	<code>\rmfamily</code>
<code>\textsf</code>	<code>\sffamily</code>
<code>\texttt</code>	<code>\ttfamily</code>

TABLE 3.2 – Correspondance entre les commandes et les déclarations de style.

3.7.1 Emphase

Une commande utile pour mettre en relief un ou des mots, est la commande :

`\emph{\langle mot(s) \rangle}`

Cette commande a l'avantage de changer de style en fonction du style courant. Si le texte est composé en romain, alors `\emph` passe en italique et *vice-versa*.

3.8 CHANGEMENT DE CORPS (TAILLE)

\LaTeX permet évidemment de changer de *corps* (de taille) de caractères. Nous avons déjà vu comment changer le corps général de composition du document dans les options de la classe chargée.

Les commandes qui permettent de changer de corps sont à utiliser avec parcimonie car le rendu visuel d'un changement de corps dans le texte est désagréable. On les réserve plutôt à la personnalisation de page de titre, de style général, de dessin, de tableau, de citation, etc.

Nous synthétisons les déclarations dans le tableau 3.3.

Le changement de corps `\Large` dans un paragraphe est désagréable à la lecture.

Le changement de corps **dans un paragraphe** est désagréable à la lecture.

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

TABLE 3.3 – Déclarations de changement de corps.

3.9 DES ENVIRONNEMENTS STRUCTURANTS

\LaTeX fournit un ensemble d’environnements qui permettent de structurer son document. Sans être exhaustif, nous allons ici en décrire quelques uns qui vous seront utiles.

3.9.1 Dissection d’un environnement

Dans un premier temps, il est important d’expliquer ce qu’est un environnement.

Définition 3.2 – Environnement

Un *environnement* est une partie du document source délimitée par les commandes :

```
\begin{<nom de l'environnement>}
\end{<nom de l'environnement>}
```

Tout comme les commandes les environnements peuvent avoir des options et des arguments. C’est à la commande d’ouverture des environnements qu’on affecte les options et les arguments.

```
\begin{<environnement>}[<options>]{<arguments>}
```

Les environnements sont aussi des *groupes* et ainsi, les changements par les déclarations (style, corps, etc.) effectuées à l’intérieur d’un environnement ne se propagent pas à l’extérieur de l’environnement comme l’illustre l’exemple suivante.

```
\begin{quote}
Voici un environnement de citation très mal utilisé, mais qui
```

```

permet de voir que \itshape la déclaration ne se propage pas
    au
    delà de l'environnement.
\end{quote}
En voilà la preuve !

```

Voici un environnement de citation très mal utilisé, mais qui permet de voir que *la déclaration ne se propage pas au delà de l'environnement*.

En voilà la preuve !

3.9.2 Les environnements quote et quotation

Comme vous avez pu apercevoir dans l'exemple précédent, \LaTeX fournit des environnements permettant de mettre en forme des citations. Il en existe deux au comportement voisin :

- l'environnement `quotation` met un alinéa en début de paragraphe et espace les paragraphes normalement ;
- l'environnement `quote` ne met pas d'alinéa en début de paragraphe et espace les paragraphes davantage.

3.9.3 Les environnements center, flushright et flushleft

Les environnements `center`, `flushright` et `flushleft` permettent respectivement de centrer du texte, de l'aligner à droite ou à gauche.

```

\begin{center}
Voici un texte centré. \LaTeX se débrouille pour couper les
    lignes
là où il faut. On peut tout de même forcer un retour avec \ la
commande \lstinline+.
\end{center}
\begin{flushleft}
\raggedright
    La composition où l'alignement est à gauche est appelée la
    composition
\emph{au fer à gauche}. Toutes les lignes sont alors alignées à
gauche. Le terme \og{}fer\fg{}fer vient de la façon dont le
compositeur

```

```
alignait les lignes sur son composteur mécanique. Regardons si
    cela
fonctionne bien.
\end{flushleft}
\begin{flushright}
On peut aussi composer les paragraphes au fer à droite. Cette
    fois ci
le texte est aligné à droite. Moins utilisé dans la plupart des
documents, cet environnement peut se révéler utile.
\end{flushright}
```

Voici un texte centré. \LaTeX se débrouille pour couper les lignes là où il faut. On peut tout de même forcer un retour avec la commande `\.`

La composition où l’alignement est à gauche est appelée la composition *au fer à gauche*. Toutes les lignes sont alors alignées à gauche. Le terme « fer » vient de la façon dont le compositeur alignait les lignes sur son composteur mécanique. Regardons si cela fonctionne bien.

On peut aussi composer les paragraphes au fer à droite. Cette fois ci le texte est aligné à droite. Moins utilisé dans la plupart des documents, cet environnement peut se révéler utile.

Pour plus de meilleurs résultats et plus de possibilités, nous conseillons l’utilisation du package `ragged2e` brièvement décrite au chapitre 8.

3.9.4 L’environnement `verbatim`

Il arrive souvent que l’on veuille composer de code informatique (\LaTeX , C++, python, etc.). \LaTeX fournit un environnement `verbatim` qui compose son contenu en style *machine à écrire*. Aucune interprétation des commande ou des caractères spéciaux n’est faite à l’intérieur de cet environnement.

```
\begin{verbatim}
L’environnement verbatim compose exactement le texte sans rien «
traiter », y compris les caractères spéciaux de  $\TeX$ , $, %
# _{ }, etc.
ou bien même les espaces qui
    ne sont pas réduits.
```

```
\end{verbatim}
```

qui produit

L'environnement `verbatim` compose exactement le texte sans rien traiter, y compris les caractères spéciaux de \TeX , `$`, `%`, `#`, `{}`, etc. ou bien même les espaces qui ne sont pas réduits.

Pour plus de possibilités nous conseillons dans cette introduction l'utilisation du package `lstlisting` brièvement décrite au chapitre 8.

3.9.5 Les listes : `itemize`, `enumerate` et `description`

Les listes sont très utiles pour structurer les documents⁷. \LaTeX en propose trois types.

La liste : `itemize`

La liste simple se compose grâce à l'environnement `itemize` comme l'illustre l'exemple suivante.

```
La liste simple :
\begin{itemize}
\item structure ;
\item àère :
\item améliore la lisibilité.
\end{itemize}
```

```
La liste simple :
– structure;
– àère :
– améliore la lisibilité.
```

Chaque élément de la liste doit commencer par la commande `\item` qui imprime, lorsque la langue est le français, un tiret.⁸

7. Et pas seulement les documents \LaTeX : sur le web et en HTML(5), les listes sont centrales.

8. Lorsque la langue est l'anglais, les éléments sont imprimés avec des *bullets* : `•`.

La liste : enumerate

L'environnement `enumerate` s'utilise exactement comme l'environnement `itemize` à la différence que lors de la composition, les éléments de la liste sont numérotés.

```
\begin{enumerate}
\item Premier élément :
  \begin{enumerate}
  \item sous élément ;
  \item sous élément ;
  \end{enumerate}

\item deuxième élément ;
\item troisième élément.
\end{enumerate}
```

-
1. Premier élément :
 - (a) sous élément;
 - (b) sous élément;
 2. deuxième élément;
 3. troisième élément.

La liste : description

L'environnement `description` permet de choisir soi-même le texte figurent au début de chaque élément de la liste au moyen de l'argument optionnel de la commande `\item` comme l'illustre l'exemple suivant.

```
\begin{description}
\item[itemize] pour faire des listes simples ;
\item[enumerate] pour faire des listes numérotées ;
\item[description] pour faire des listes dont chaque élément
  débute
  par le texte en gras de son choix.
\end{description}
```

itemize pour faire des listes simples ;
enumerate pour faire des listes numérotées ;
description pour faire des listes dont chaque élément débute par le texte en

gras de son choix.

3.9.6 Les tableaux

\LaTeX permet de composer des tableaux grâce à l’environnement `tabular`. La composition des tableaux n’est pas toujours un exercice facile avec \LaTeX mais avec un peu d’effort, on peut quasiment tout faire. Nous nous focaliserons ici sur la présentation de l’environnement `tabular`.

Pour se familiariser avec les mécanismes, commençons par décortiquer un exemple simple.

```
\begin{tabular}{lcr}
& colonne centrée & on aligne à droite \\
Ligne 1 & ici & là \\
Ligne deux & là-bas & ailleurs
\end{tabular}
```

	colonne centrée	on aligne à droite
Ligne 1	ici	là
Ligne deux	là-bas	ailleurs

L’environnement `tabular` comporte un argument. Ici, il comporte trois lettres. Le nombre de lettre indique le nombre de colonnes du tableau que l’on souhaite composer. La type de lettre indique l’alignement de la colonne :

l (LEFT) : alignement à gauche de la colonne ;

c (CENTER) : centrage de la colonne ;

r (RIGHT) : alignement à droite de la colonne.

Il existe d’autres type de colonnes que nous allons présenter via quelques exemples. Le nombre de lettres dans l’argument de `tabular` détermine le nombre de colonnes du tableau.

À l’intérieur de l’environnement, les lignes sont terminées par `\\` (sauf la dernière) et les colonnes sont séparées par `&`.

On peut aussi matérialiser les séparations de lignes et de colonnes dans l’argument et à l’intérieur de l’environnement comme le montre l’exemple suivant.

```
\begin{tabular}{[ l | c | r ]}
\hline
& colonne centrée & on aligne à droite \\
\hline
Ligne 1 & ici & là \\
\end{tabular}
```

```

\hline
Ligne deux & là-bas & ailleurs\\
\hline
\end{tabular}

```

	colonne centrée	on aligne à droite
Ligne 1	ici	là
Ligne deux	là-bas	ailleurs

Comme illustré, les lignes tracées entre les colonnes sont obtenues en plaçant des | entre les lettres définissant les alignements des colonnes dans l'argument de l'environnement tabular (ces lignes s'entendant sur toute la hauteur du tableau).

Les lignes horizontales sont obtenues par la commande \hline après la commande \\ placée entre chaque ligne du tableau.

L'argument p — Un autre type de colonne existe qui peut être très utile : `p{taille}`. Ceci permet de déclarer une colonne d'une largeur *taille*. À l'intérieur des cases de cette colonne, le contenu sera composé comme un paragraphe, et le texte sera aligné à gauche.

Déclaration de plusieurs même styles — Pour les grands tableaux, et lorsque les colonnes ont le même style alors, on peut factoriser la déclaration grâce à la commande `*{nombre}{format}`

Voici un petit exemple qui illustre les deux derniers arguments.

```

\begin{tabular}[p{5cm}]{*2}{c|}}
\cline{2-3}
\multicolumn{1}{c []}{} & \multicolumn{2}{c []}{Commande} \\
\cline{2-3}
\multicolumn{1}{c []}{} & Prix & Quantité \\
\hline
\bfseries Tableaux & 180 \euro & 3 \\
\hline
\bfseries Craies (x50) & 10 \euro & 5 \\
\hline
\bfseries Brosses & 5 \euro & 3 \\
\hline
\end{tabular}

```

	Commande	
	Prix	Quantité
Tableaux	180 €	3
Craies (x50)	10 €	5
Brosses	5 €	3

3.10 INCLURE DES IMAGES

Dans les préambules présentés dans ce document, on charge le package `graphicx` :

```
\usepackage{graphicx}
```

préambule

Celui-ci permet d'avoir une commande supplémentaire très importante :

```
\includegraphics[<options>]{<nom de l'image>}
```

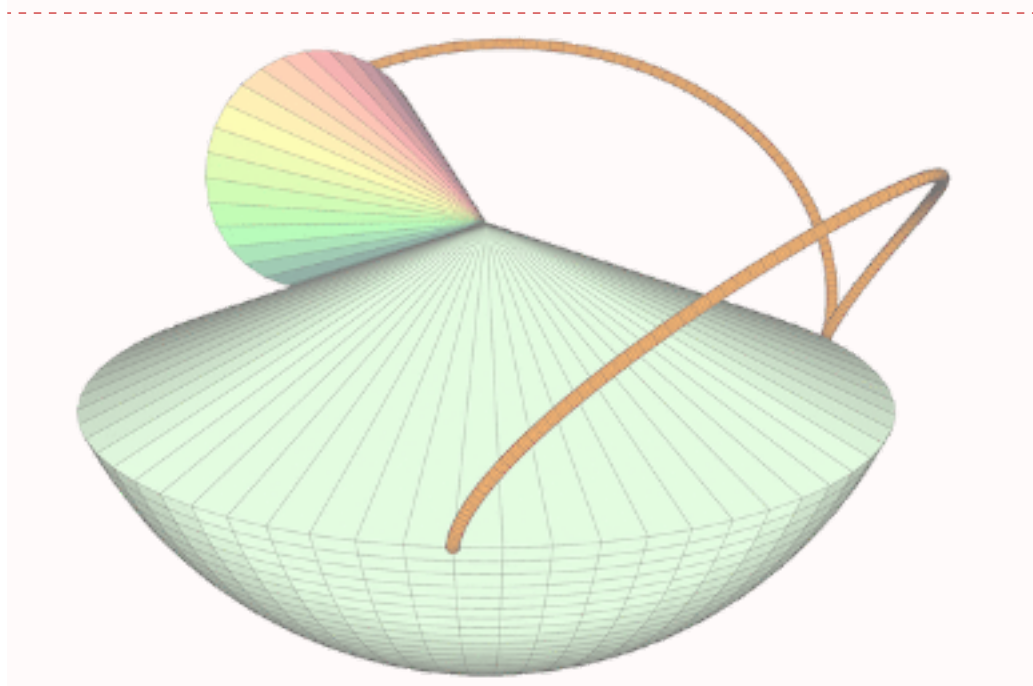
qui permet⁹ d'inclure des images au format, entre autres, PDF, png, ou bien JPEG. On pourra se rapporter à la documentation du package [9] pour plus de détail. Notons que ce package permet plus de choses que la seule inclusion d'image.

Si je dispose de l'image `logo.png` à la racine de mon document de travail¹⁰, alors il suffit d'inclure l'image comme ceci :

```
\includegraphics{logo.png}
```

9. avec la compilation `pdflatex...`

10. Il est préférable d'organiser son répertoire de travail, et d'y ajouter, par exemple, un sous-répertoire pour y mettre les images.



De nombreuses options existent pour cette commande. Nous ne présenterons ici que les plus courantes. Ces options sont à utilisées sous la forme clé=valeur, comme le montre les exemples suivants.

angle : angle de rotation de l'image en degrés.

width : largeur (avec unité) de l'image voulue. La transformation se faisant en préservant le rapport entre largeur et hauteur.

height : hauteur (avec unité) de l'image voulue. La transformation se faisant en préservant le rapport entre largeur et hauteur.

scale : facteur d'échelle.

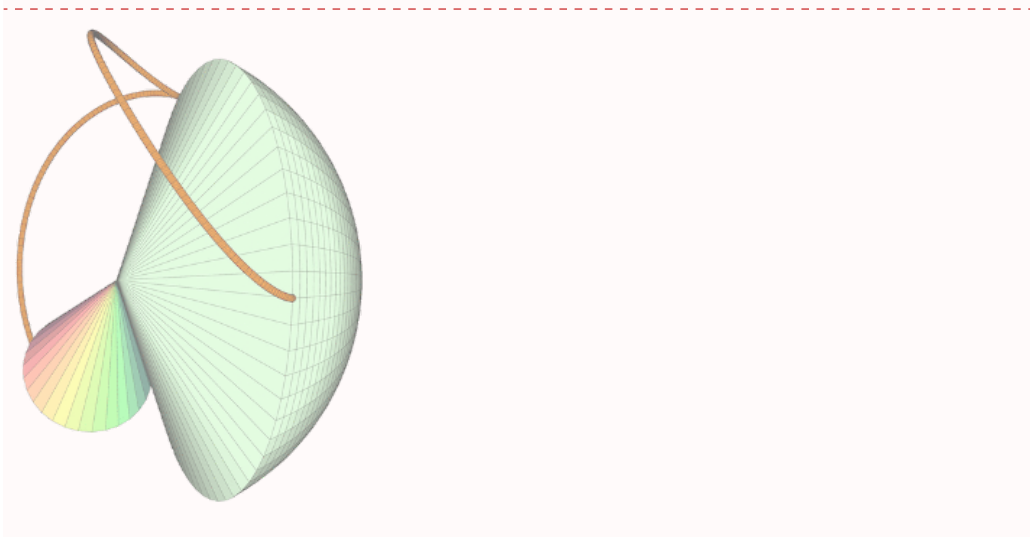
page : numéro de la page du PDF à inclure (lorsqu'il s'agit d'un fichier PDF multi-pages).

Dans l'exemple suivant, on introduit la commande

`\linewidth`

qui donne la largeur de la page (largeur de composition, c'est-à-dire, là où il y a du texte).

```
\includegraphics[width=0.5\linewidth,angle=90]{logo.png}% on veut
    une figure de
                                % largeur la moitié de la largeur des
                                % lignes de texte
```



Notons que l'alignement de l'image incluse directement dans du texte se fait sur la ligne de référence de composition comme l'illustre l'exemple suivant.

Du texte qui permet de constater l'alignement
de l'image~\includegraphics[scale=0.1]{logo.png}.

Du texte qui permet de constater l'alignement de l'image  .

On utilisera souvent cette commande d'inclusion dans l'environnement figure décrite en section 4.3 qui permet d'inclure une figure et d'y faire référence.

FLOTTANTS, RÉFÉRENCES CROISÉES, BIBLIOGRAPHIES ET INDEX

Dans ce chapitre, nous abordons un des aspects très puissant de \LaTeX . En effet, avec l'utilisation de code, de balises, on peut faire référence à des objets (sections, tableaux, images, etc.) sans se soucier de l'endroit où ces objets se trouveront puisque le système va déterminer automatiquement la page, le numéro, etc. de l'objet. Des mécanismes automatiques de gestion de bibliographies et d'index existent aussi. Tous ces outils permettent de produire des documents toujours cohérents, même après des restructurations importantes (le liens étant mis à jour à chaque compilation), avec des bibliographies générées automatiquement suivant les ouvrages cités, des index produits automatiquement, etc. Enfin, ces mécanismes sont suffisamment puissants pour que le facteur d'échelle ne soit pas un problème, permettant ainsi de produire d'énormes documents.

4.1 RÉFÉRENCES CROISÉES

\LaTeX dispose d'un mécanisme de *références croisées*. Cela permet, dans le texte courant, de faire référence à des parties du document (sections, chapitres, parties), des figures, des tableaux (via les mécanismes de flottants décrit dans la section suivante), des équations, des théorèmes, etc.

Montrons un exemple pour fixer les idées.

```
\section{Titre de section}\label{sec:txtderef}
```

Voici la section~\ref{sec:txtderef} à laquelle on peut faire référence.

```
\begin{equation}\label{eq:integrale}
\int_0^1 x^2 \mathrm{d}x
\end{equation}
```

Calculez l'intégrale~\ref{eq:integrale}.

4.2 TITRE DE SECTION

Voici la section 4.2 à laquelle on peut faire référence.

$$\int_0^1 x^2 dx \tag{4.1}$$

Calculez l'intégrale 4.1.

Le mécanisme de référence croisée repose donc sur un couple de commandes :

```
\label{(textedelabel)}
```

et

```
\ref{(textedelabel)}
```

Le fonctionnement est simple, on définit une *chaîne de caractère* via la commande `\label` près ou dans un élément numéroté de \LaTeX (section, théorème, équation, figure, table, etc.), et on fait référence à cet élément par la commande `\ref` qui appelle la même chaîne de caractère. Il est d'usage, et c'est une bonne habitude, de préfixer la chaîne de caractère par une indication du type de l'élément à référencer. Dans notre exemple, il s'agit de `sec` : pour section, et de `eq` : pour équation.

Le résultat final est qu'en lieu et place de la commande `\ref`, il apparaît le numéro de l'élément référencé.¹ Pour que le résultat s'affiche correctement, il faudra effectuer deux compilations successives, car les références passent par l'écriture d'un fichier auxiliaire qui doit être lu ensuite lors de la compilation. Lors de la première compilation, le résultat temporaire sera constitué de deux points d'interrogations : `??`. Ne vous en étonnez pas.

Une autre commande permet de faire référence au numéro de page contenant l'élément auquel on fait référence. Il s'agit de la commande

1. Ces éléments référençables sont associés à des compteurs internes et c'est la valeur de ce compteur qui est affichée.

`\pageref{<textedelabel>}`

Un des avantages énorme de ce mécanisme est que même lorsque l'on déplace le contenu ou que l'on restructure notre document, les références continuent d'être correctes.

4.3 FLOTTANTS : table ET figure

Dans la composition de document, le concept de *flottant* est très important. L'insertion de certains types d'objets comme les figures ou les images peuvent poser quelques problèmes lors de la composition des pages : il est important d'avoir un certain taux de remplissage de la page, de ne pas couper la page n'importe où, etc.

Pour résoudre ce problème, \LaTeX dispose de la notion de *flottants*, c'est-à-dire un objet qui a une certaine fluctuation de position mais auquel il est fait *référence* dans le texte courant.

Nous présenterons ici deux environnements flottants : `table` et `figure` pour les tableaux et les figures et qui permettent d'inclure ces éléments en leur donnant un titre et en les numérotant, et d'y faire référence.

Décortiquons un exemple basique de l'environnement `table` pour mieux comprendre :

```

\begin{table}[!htbp]
\centering
\begin{tabular}{|l|l|}
\hline
Élément 1 & Élément 2 \\
\hline
Élément 3 & Élément 4 \\
\hline
\end{tabular}
\caption{Exemple 'd'environnement \lstinline+table+}
\label{tab:exemple}
\end{table}

```

qui produit le tableau 4.1.

Les arguments optionnels des environnements flottants sont très importants et doivent être bien compris. La suite des caractères entre crochets déterminent l'ordre de préférence du placement du tableau :

Élément 1	Élément 2
Élément 3	Élément 4

TABLE 4.1 – Exemple d’environnement table

! : demande à \LaTeX de faire tout son possible pour respecter l’ordre indiqué ensuite;

h (HERE) : ici, si possible;

t (TOP) : en haut d’une page (la courante ou la suivante);

b (BOTTOM) : en bas d’une page (la courante ou la suivante);

p (PAGE OF FLOATS) : sur une page spéciale ne contenant pas de texte mais uniquement des tableaux et des figures.

Continuons la description du code précédent.

- L’ordre de préférence par défaut est tbp.
- La commande `\centering` est là pour centrer le tableau. Une fois n’est pas coutume, elle est ici préférable à l’environnement `center` car celui-ci produirait une marge trop importante entre le tableau et sa légende.
- La commande `\caption{}` permet de donner une légende au tableau.
- La commande `\label{}`, qui doit être obligatoirement placée après la commande `\caption`, permet de faire référence à l’élément via le mécanisme des références croisées vu en section 4.1 et les commandes `\ref` ou `\pageref`.

Ce fonctionnement de flottant n’est souvent pas bien compris, et spontanément, on souhaite exactement choisir l’emplacement de notre figure ou de notre tableau. Avec l’habitude, il est bien plus naturel de faire référence à l’élément flottant dans le texte courant et de laisser l’algorithme d’optimisation de \LaTeX produire les pages.

L’environnement `figure` s’utilise exactement de la même façon.

4.3.1 Listes des tableaux et des figures

Il est possible de faire figurer dans votre document la liste des tableaux au moyen de la commande `\listoftables` et la liste des figures au moyen de la commande `\listoffigures`.

De la même manière que pour les références pour lesquelles \LaTeX écrit des fichiers auxiliaires lors de la compilation, pour que ces listes soient à jour, il faut effectuer deux compilations successives. Les fichiers auxiliaires sont les fichiers `.lot` et `.lof`.

```
\listoffigures
```

TABLE DES FIGURES

3.1	Le fichier <code>premier.tex</code> dans <code>T_EXstudio</code>	19
3.2	Fichier PDF produit par la compilation <code>pdflatex</code>	20
6.1	Exemple de diapositive avec <code>beamer</code> avec le thème <code>Rochester</code> et le thème de couleur <code>seagull</code>	86
6.2	Page de titre avec <code>beamer</code> avec le thème <code>Rochester</code> et le thème de couleur <code>seagull</code>	87
6.3	La table des matières avec le thème <code>Rochester</code> et le thème de couleur <code>seagull</code>	88
6.4	Les différents blocs avec le thème <code>Rochester</code> et le thème de couleur <code>seagull</code>	89
7.1	Figure illustrant les courbes de Bézier cubiques.	99
8.1	Illustration de l'utilisation du package <code>refcheck</code>	104
8.2	Résultat de l'utilisation du package <code>todonotes</code>	114
8.3	Exemple de composition d'un exercice, de sa corrections et des annotations en utilisant le package <code>exosheet</code>	117
8.4	Résultat d'un simple code Python avec le package <code>listings</code> . .	118

4.4 BIBLIOGRAPHIE – biblalex

Un travail scientifique utilise de nombreuses références bibliographiques. `LATEX` offre des outils très performants pour la gestion des bibliographies. Nous en présenterons ici l'utilisation de l'extension `biblalex`.

La présentation faite ici est bien plus qu'inspirée de la présentation [39]. Pour encore plus de détails, on pourra consulter [40]. Pour la documentation de `biblalex`, on consultera [25].

L'outil, qui est en fait un ensemble d'outils, nous permet de :

- gérer et organiser ses références ;

- citer de manière cohérente et suivant une norme ;
- liste de manière exhaustive, selon un ordre précis.

Nous nous limiterons à présenter rapidement les fonctionnalités principales. Il ne faudra pas hésiter à aller voir la documentation de `biblatex` pour avoir les informations exhaustives.

4.4.1 Stocker sa base bibliographique

Avec \LaTeX , la base bibliographique se stocke dans un fichier d'extension `.bib`. Pour clarifier, appelons notre fichier `mabibliographie.bib`.

Ce fichier contient des *entrées bibliographiques* qui ressembleront à l'exemple suivant que nous allons encore une fois décortiquer.

```
@book{Orwell1984,
  asin = {0881030368},
  author = {Orwell, George},
  dewey = {823.912},
  ean = {9780881030365},
  edition = {Centennial.},
  isbn = {0881030368},
  publisher = {Tandem Library},
  title = {1984},
  year = 1950
}
```

À l'image de l'exemple ci-dessus, les entrées se décomposent comme suit :

- un type précédé d'un `@`, ici `@book` pour un livre ;
- une clé, après la première accolade et avant la virgule, ici `Orwell1984`, c'est cette clé qui permettra de faire référence à l'entrée bibliographique dans le texte \LaTeX ;
- des champs sous forme : $\langle \textit{nom du champ} \rangle = \langle \textit{valeur} \rangle$, séparés par des virgules.

Quelques types d'entrées

Voici quelques types de base :

@article : comme son nom l'indique ;

@book : livre avec un·e ou plusieurs auteur·e·s principaux ;

@collection : livre composé de plusieurs articles d'auteur·e·s distinct·e·s ;

@manual : pour les manuels ;

@reference : ouvrage de référence, tel que dictionnaire ou encyclopédie ;

@online : ressource en ligne ;

@report : rapport technique ;

@patent : brevet industriel ;

@periodical : numéro particulier d'un périodique ;

@proceedings : actes de colloque ;

@thesis : thèse de doctorat ou mémoire de master.

Champs de personne

Il existe beaucoup de champs dits de personne qui permettent de préciser le rôle qu'ont pu avoir les personnes qui ont participé à l'ouvrage. En voici quelques uns (il en existe beaucoup d'autres) :

author : auteur·trice(s) de l'œuvre ;

bookauthor : auteur·trice(s) du livre dans lequel l'œuvre est insérée ;

commentator : auteur·trice(s) des commentaires ;

editor : éditeur·trice(s) scientifique(s) ;

introduction : auteur·trice(s) de l'introduction ;

translator : traducteur·trice(s).

La valeur des ces champs se formate de la façon suivante.

- Le mot clé and permet de séparer plusieurs auteurs et autrices.
- On écrit les noms et prénoms comme ceci : $\langle Nom \rangle$, $\langle Prénom1 \rangle$ $\langle Prénom2 \rangle$.
- Pour les auteurs collectifs, il faut utiliser les accolades, par exemple :

```
Author = {{Centre National de la Recherche Scientifique}}
```

Champs de titre

En voici trois très important :

title : Titre de l'œuvre.

subtitle : Sous-titre de l'œuvre.

journaltitle : Titre d'un périodique.

Champs de description éditorial

Voici quelques champs utiles :

date : Date de publication.

edition : Numéro d'édition si plusieurs éditions existent. location Lieu de publication.

number : Numéro d'un périodique ou numéro au sein d'une collection.

pages : Pages de l'article ou de la partie du livre étudiée.

publisher : Éditeur commercial.

url : Url (adresse électronique) d'une publication en ligne.

volume : Volume dans une œuvre en plusieurs volumes. Volume d'une revue.

volumes : Nombre de volumes dans une œuvres en plusieurs volumes.

Constituer et gérer sa base bibliographique

Plutôt que d'éditer son fichier `.bib` à la main, il existe des logiciels dédiés qui présentent de nombreux avantages :

- ils permettent d'éviter un certain nombre d'erreurs dans la composition du fichier `.bib`;
- ils permettent de faciliter les recherches dans notre base bibliographique ;
- permettent de rechercher automatiquement dans certaines bases bibliographiques en ligne.

Parmi ces logiciels, nous mentionnons ici :

- Zotero est un logiciel multiplateforme pour la gestion de base bibliographique et n'est pas spécifique à bibtex ou biblatex : <https://retorque.re/zotero-better-bibtex/> ;
- JabRef est un logiciel multiplateforme pour la gestion de base bibliographique spécifique à bibtex ou biblatex : <https://www.jabref.org/> ;
- BibDesk est un logiciel pour Mac OSX spécifique à bibtex ou biblatex : <https://bibdesk.sourceforge.io/>.

MathSciNet et zbMATH — Comme tout, internet permet de trouver les entrées bibliographiques déjà formatées pour biblatex. Cependant pour les ouvrages, et plus généralement les publications mathématiques, depuis l'université, nous avons accès à la base de donnée MathSciNet (<https://mathscinet.ams.org/>) qui permet d'avoir les exportations sous le format bibtex des ouvrages. Le système zbMBTH (<https://zbmath.org>) équivalent mais ouvert et gratuit est aussi très pratique et fonctionne comme MathScinet.

4.4.2 Charger le package et ajouter la bibliographie

Dans le préambule de notre document \LaTeX il nous faut ajouter les lignes suivantes :

```
\usepackage[<options>]{biblatex}
\addbibresource{<nom>.bib}
```

Il est souvent commode de mettre le fichier `.bib` dans à côté du document \LaTeX . Il est cependant possible de centraliser dans un répertoire accessible à \LaTeX : `texmfhome` → `bibtex` → `bib` où `texmfhome` est un certain dossier qui dépend du système et de l'installation mais qui s'obtient avec le commande

```
user $> kpsewhich --var-value TEXMFHOME
```

Dans les options au chargement de `biblatex`, on peut personnaliser le style de notre bibliographie. Nous renvoyons à [39] pour les différentes options.

4.4.3 Citer

Pour citer une référence, il faut utiliser la commande suivante :

```
\cite[<prénote>][<postnote>]{<clef>}
```

Montrons dans l'exemple suivant le résultat d'une telle commande.

```
\cite[test][coucou]{Rouquette2019}
```

```
[test 39, coucou]
```

Il arrive souvent que l'on veuille citer plusieurs références qui traitent du même sujet. On utilisera alors la commande suivante :

```
\cites(<prénote globale>)(<postnote global>)[<prenote 1>][<postnote 1>]{<clef 1>}
[<prenote 2>][<postnote 2>]{<clef 2>} etc.
```

Par l'exemple, cela donne :

```
On consultera les trois textes fondamentaux pour débiter avec
\LaTeX{~}\cites(Des PDF)(importants)[Denis][Bitouzé]{Denis}[
  Babafou]{babafou}[biblatex][biber]{Rouquette2019}.
```

On consultera les trois textes fondamentaux pour débiter avec \LaTeX [Des PDF Denis 6, Bitouzé, 2, Babafou, biblatex 39, biber, importants].

En général, on utilise :

- `\prenote` pour un texte à afficher avant la référence du type de « Voir » ;
- `\postnote` pour la page. En effet, lorsque l'argument est réduit à un numéro, alors l'abréviation de page est ajouté.²

```
\cite[cf.][]{Denis}. Pour la bibliographie
\cite[Voir][49]{Rouquette2019}, ou bien \cite[3]{babafou}.
```

[cf. 6]. Pour la bibliographie [Voir 39, p. 49], ou bien [2, p. 3].

Les potentialités de citations sont très importantes. Nous renvoyons aux références pour explorer cela, notamment les commandes `\autocite` et `\autocites` qui sont à préférer et qui ont la même syntaxe que `\cite` et `\cites`.

```
\autocite{Rouquette2019}
```

[39]

Ces commandes `\autocite` et `\autocites` adaptent le format d'affichage au style choisi et au contexte.

Notons tout de même que l'on peut citer uniquement certaines informations. `biblatex` fournit un ensemble de commandes de citations qui peuvent être très utiles :

- `\citeauthor` : pour l'auteur (ou l'éditeur ou le traducteur si pas d'auteur) ;
- `\citetitle` : pour le titre ou le titre abrégé ;
- `\citeyear` : pour l'année ;
- `\citedate` : pour la date ;
- `\fullcite` : pour la citation complète, sans aucune abréviation ;
- `\footfullcite` : pour la citation complète, en notes de bas de page ;
- `\nocite` : ne cite pas l'entrée, mais l'ajoute à la bibliographie finale. Si l'argument est un *, cette commande ajoute toutes les entrées de la base de données.

2. Et lorsque qu'il n'y a qu'un nombre en argument optionel alors, celui-ci est traité comme une `\postnote` à laquelle est ajouté l'abréviation de page.

4.4.4 Établir la bibliographie finale

Nous venons de voir comment créer et gérer son fichier `.bib` contenant notre base bibliographique, ainsi que le procédé pour citer les références au long du texte de notre document \LaTeX .

Maintenant, il nous faut voir comment composer la liste des références citées dans le document. Cela se fait avec la commande

```
\printbibliography[⟨options⟩]
```

qui va afficher toutes les entrées citées.

Là encore les options sont nombreuses (locales ou au chargement du package), et nous renvoyons à la documentation de l'extension `biblatex` pour plus de détails, mais pour montrer les possibilités de `biblatex`, arrêtons nous sur quelques unes.

Styles

Il existe de très nombreux styles de bibliographie. Nous renvoyons à [25, p. 68] pour la listes des styles disponibles. C'est au chargement du package que celui se choisit. Le style se choisit comme ceci `\usepackage[style=<valeur>]{biblatex}`.

Trier la bibliographie

Au chargement du package toujours, on peut modifier le tri par défaut du style choisi. Ceci se fait avec l'option `sorting=` :

- `nyt` qui permet un tri par auteur, année et titre ;
- `nty` qui permet un tri par auteur, titre et année ;
- `ynt` qui permet un tri par année, et titre ;
- et bien d'autres possibilités encore.

Options locales

Comme montré plus haut la commande `\printbibliography` possède des options. Parmi ces options, certaines permettent de personnaliser la composition de la bibliographie.

Apparence — Suivant la classe du document, la commande `\printbibliography` va générer un chapitre ou une section, non numérotée, dont le titre sera, en français *Bibliographie*. Ceci peut être modifié par les options locales.

Pour changer le niveau de titre de la bibliographie, il faudra utiliser la commande `heading=<xxx>`, ou `<xxx>` peut être :

bibliography : (par défaut) qui introduit donc, suivant la classe du document, un `\chapter*` ou une `\section*`;

subbibliography : qui produira, toujours suivant la classe du document, une `\section*` ou une `\subsection*`;

bibintoc : qui produira un `\chapter*` ou une `\section*` mais qui sera ajouté à la table des matières;

subbibintoc : qui produira, toujours suivant la classe du document, une `\section*` ou une `\subsection*` ajouté à la table des matières;

bibnumbered : qui introduit un `\chapter` ou une `\section` numéroté(e);

subbibnumbered : qui produira, toujours suivant la classe du document, une `\section` ou une `\subsection` numérotée.

Pour changer le titre de la bibliographie, il faudra utiliser l'option `title={xxx}`, où `<xxx>` est ce qui va être placé dans la commande de *sectionnement*.

```
\printbibliography[heading=subbibliography,title={Ma
bibliographie}]
```

MA BIBLIOGRAPHIE

- [1] Jacques ANDRÉ. *Petites leçons de typographie*. <http://jacques-andre.fr/faqtypo/lessons.pdf>. 2017.
- [2] Marc BAUDOIN. *Apprends L^AT_EX!* 2012. URL : <https://www.ctan.org/tex-archive/info/apprends-latex/>.
- [3] Claudio BECCARI et Enrico GREGORIO. *The imakeidx package. A package for producing multiple indexes*. Version 1.3e. 15 oct. 2016. URL : <https://ctan.org/pkg/imakeidx>.
- [4] Denis BITOUZÉ. *Conférence 1 : les bases de L^AT_EX*. 2020. URL : <https://gte.univ-littoral.fr/Members/denis-bitouze/pub/latex/diapositives-cours-d/conference-n-1/@@download/file/en-ligne1.pdf>.
- [5] Denis BITOUZÉ. *Site web*. 2020. URL : <http://gte.univ-littoral.fr/members/dbitouze/pub>.
- [6] Denis BITOUZÉ et Jean-Côme CHARPENTIER. *Introduction à L^AT_EX*. Pearson, juill. 2010. ISBN : 2-7440-7451-9. URL : <http://www.pearson.fr/livre/?GCOI=27440100259650>.

- [7] Rogério BRITO et Peter WILLIAMS. *The algorithms package. A suite of tools for typesetting algorithms in pseudo-code*. Version 0.1. 9 oct. 2018. URL : <https://ctan.org/pkg/algorithms>.
- [8] David CARLISLE, David KASTRUP et THE L^AT_EX PROJECT TEAM. *The longtable package. Allow tables to flow over page boundaries*. Version 4.19. 1^{er} nov. 2023. URL : <https://ctan.org/pkg/longtable>.
- [9] David CARLISLE et THE L^AT_EX PROJECT TEAM. *The graphicx package. Enhanced support for graphics*. Version 1.2d. 12 nov. 2021. URL : <https://ctan.org/pkg/graphicx>.
- [10] David CARLISLE et THE L^AT_EX PROJECT TEAM. *The tabularx package. Tabulars with adjustable-width columns*. Version 2.11c. 1^{er} nov. 2023. URL : <https://ctan.org/pkg/tabularx>.
- [11] Daniel ELS et Simon FEAR. *The booktabs package. Publication quality tables in L^AT_EX*. Version 1.61803398. 14 jan. 2020. URL : <https://ctan.org/pkg/booktabs>.
- [12] Christophe FIORIO et Ivan KOKAN. *The algorithm2e package. Floating algorithm environment with algorithmic keywords*. Version 5.2. 6 avr. 2020. URL : <https://ctan.org/pkg/algorithm2e>.
- [13] Daniel FLIPO. *The babel-french package. Babel contributed support for French*. Version 3.5q. 9 mars 2023. URL : <https://ctan.org/pkg/babel-french>.
- [14] Yvon HENEL. *The tdsfrmath package. Macros for French teachers of mathematics*. Version 1.3. 24 juin 2016. URL : <https://ctan.org/pkg/tdsfrmath>.
- [15] Jobst HOFFMANN, Brooks MOSES et Carsten HEINZ. *The listings package. Typeset source code listings using L^AT_EX*. Version 1.9. 6 oct. 2023. URL : <https://ctan.org/pkg/listings>.
- [16] Morten HØGHOLM et al. *The mathtools package. Mathematical tools to use with amsmath*. Version 1.29. 30 juin 2022. URL : <https://ctan.org/pkg/mathtools>.
- [17] Oliver IRWIN. *The frpseudocode package. French translation for the algorithmicx package*. Version 0.3.0. 10 août 2020. URL : <https://ctan.org/pkg/frpseudocode>.
- [18] Szász JÁNOS. *The algorithmicx package. The algorithmic style you always wanted*. 24 juin 2016. URL : <https://ctan.org/pkg/algorithmicx>.

- [19] Alan JEFFREY, Frank MITTELBACH et THE L^AT_EX PROJECT TEAM. *The inputenc package. Accept different input encodings*. Version 1.3d. 1^{er} juin 2021. URL : <https://ctan.org/pkg/inputenc>.
- [20] Roger KEHR et Joachim SCHROD. *The xindy package. A general-purpose index processor*. Version 2.5.1. 18 avr. 2017. URL : <https://ctan.org/pkg/xindy>.
- [21] Uwe KERN et THE L^AT_EX PROJECT TEAM. *The xcolor package. Driver-independent color extensions for L^AT_EX and pdfLaTeX*. Version 2.14. 12 juin 2022. URL : <https://ctan.org/pkg/xcolor>.
- [22] Donald E. KNUTH. *Le T_EXbook : composition informatique*. Trad. par Jean-Côme CHARPENTIER. Vuibert informatique. Vuibert, 2003. ISBN : 9782711748198.
- [23] *L'association GUTenberg*. 2024. URL : <https://gutenberg-asso.fr>.
- [24] Leslie LAMPORT. *L^AT_EX : a document preparation system*. Anglais. Second. Addison-Wesley, 1994. ISBN : 0-201-52983-1. URL : <http://www.informit.com/store/product.aspx?isbn=0201529831>.
- [25] Philipp LEHMAN et Philip KIME. *The BibLaTeX package. Sophisticated Bibliographies in L^AT_EX*. Version 3.19. 6 mars 2023. URL : <https://ctan.org/pkg/biblatex>.
- [26] Javier Bezos LÓPEZ. *The enumitem package. Control layout of itemize, enumerate, description*. Version 3.9. 22 juin 2019. URL : <https://ctan.org/pkg/enumitem>.
- [27] Javier Bezos LÓPEZ et Johannes L. BRAAMS. *The babel package. Multilingual support for L^AT_EX, LuaL^AT_EX, XeL^AT_EX, and Plain T_EX*. Version 3.96. 25 oct. 2023. URL : <https://ctan.org/pkg/babel>.
- [28] Jianrui LYU. *The tabularray package. Typeset tabulars and arrays with L^AT_EX3*. Version 2023A. 1^{er} mars 2023. URL : <https://ctan.org/pkg/tabularray>.
- [29] Christian MATT. *The algpseudocodex package. Package for typesetting pseudocode*. Version 1.1.2. 19 avr. 2023. URL : <https://ctan.org/pkg/algpseudocodex>.
- [30] Wolfgang MAY et Andreas SCHEDLER. *The ntheorem package. Enhanced theorem environment*. Version 1.33. 18 avr. 2017. URL : <https://ctan.org/pkg/ntheorem>.
- [31] Henrik Skov MIDTIBY. *The todonotes package. Marking things to do in a L^AT_EX document*. Version 1.1.6. 31 jan. 2023. URL : <https://ctan.org/pkg/todonotes>.

- [32] Vedran MILETIĆ et al. *The beamer package. A L^AT_EX class for producing presentations and slides*. Version 3.70. 15 mai 2023. URL : <https://ctan.org/pkg/beamer>.
- [33] Antoine MISSIER. *The exesheet package. Typesetting exercise or exam sheets*. Version 2.4. 29 oct. 2023. URL : <https://ctan.org/pkg/exesheet>.
- [34] Frank MITTELBACH et THE L^AT_EX PROJECT TEAM. *The array package. Extending the array and tabular environments*. Version 2.5g. 1^{er} nov. 2023. URL : <https://ctan.org/pkg/array>.
- [35] Oleg V. MOTYGIN. *The refcheck package. Check references (in figures, table, equations, etc)*. Version 1.9.1. 24 juin 2016. URL : <https://ctan.org/pkg/refcheck>.
- [36] T_EXnique. 2020. URL : <https://texnique.fr/osqa/>.
- [37] Heiko OBERDIEK et al. *The hyperref package. Extensive support for hyper-text in L^AT_EX*. Version 7.01d. 31 oct. 2023. URL : <https://ctan.org/pkg/hyperref>.
- [38] OPEN-FREAX. *Personnaliser ses listes L^AT_EX avec enumitem*. 2013. URL : <https://open-freax.fr/listes-latex-enumitem/> (visité le 25/09/2020).
- [39] Maïeul ROUQUETTE. *BibL^AT_EX and biber*. 2019. URL : <https://geekographie.maieul.net/211>.
- [40] Maïeul ROUQUETTE, Enimie ROUQUETTE et Brendan CHABANNES. *(Xe)L^AT_EX appliqué aux sciences humaines*. Tampere : Atramenta. 268 p. ISBN : 978-952-273-073-2.
- [41] Ulrich M. SCHWARZ et Yukai CHOU. *The thmtools package. Extensions to theorem environments*. Version 76. 4 mai 2023. URL : <https://ctan.org/pkg/thmtools>.
- [42] T_EX Stackexchange. 2020. URL : <https://tex.stackexchange.com>.
- [43] Thomas F. STURM. *The tcolorbox package. Coloured boxes, for L^AT_EX examples and theorems, etc*. Version 6.1.0. 26 sept. 2023. URL : <https://ctan.org/pkg/tcolorbox>.
- [44] THE L^AT_EX PROJECT TEAM. *The amsmath package. AMS mathematical facilities for L^AT_EX*. Version 2.17o. 9 juin 2023. URL : <https://ctan.org/pkg/amsmath>.
- [45] THE L^AT_EX PROJECT TEAM. *The fontenc package. Standard package for selecting font encodings*. 24 juin 2016. URL : <https://ctan.org/pkg/fontenc>.

- [46] THE L^AT_EX PROJECT TEAM. *The makeidx package. Standard L^AT_EX package for creating indexes.* Version 1.0m. 9 juin 2023. URL : <https://ctan.org/pkg/makeidx>.
- [47] Gérard TISSEAU et Jacques DUMA. *Tikz pour l' impatient.* <http://math.et.info.free.fr/TikZ/index.html>. 2017.
- [48] Hideo UMEKI et David CARLISLE. *The geometry package. Flexible and complete interface to document dimensions.* Version 5.9. 31 oct. 2021. URL : <https://ctan.org/pkg/geometry>.
- [49] Joseph WRIGHT. *The siunitx package. A comprehensive (SI) units package.* Version 3.3.7. 7 oct. 2023. URL : <https://ctan.org/pkg/siunitx>.

Filtrage — Grâce aux options locales, on peut filtrer les entrées bibliographiques à afficher. Les possibilités principales sont les suivantes :

keyword=<xxx> : qui permet de sélectionner les entrées ayant <xxx> comme mot-clé enregistré dans le champ keywords dans le fichier .bib ;

type=<xxx> : qui permet de sélectionner les entrées par type (article,book, etc.);

et bien d'autres choses encore nous renvoyons à la documentation de l'extension biblatex [25] pour plus de détails.

```
\chapter*{Bibliographie}
\printbibliography[heading=subbibliography,title={Les livres},
type=book]
```

BIBLIOGRAPHIE

LES LIVRES

- [1] Jacques ANDRÉ. *Petites leçons de typographie*. <http://jacques-andre.fr/faqtypo/lessons.pdf>. 2017.
 - [2] Marc BAUDOIN. *Apprends L^AT_EX!* 2012. URL : <https://www.ctan.org/tex-archive/info/apprends-latex/>.
 - [6] Denis BITOUZÉ et Jean-Côme CHARPENTIER. *Introduction à L^AT_EX*. Pearson, juill. 2010. ISBN : 2-7440-7451-9. URL : <http://www.pearson.fr/livre/?GCOI=27440100259650>.
 - [22] Donald E. KNUTH. *Le T_EXbook : composition informatique*. Trad. par Jean-Côme CHARPENTIER. Vuibert informatique. Vuibert, 2003. ISBN : 9782711748198.
 - [24] Leslie LAMPORT. *L^AT_EX : a document preparation system*. Anglais. Second. Addison-Wesley, 1994. ISBN : 0-201-52983-1. URL : <http://www.informit.com/store/product.aspx?isbn=0201529831>.
 - [40] Maïeul ROUQUETTE, Enimie ROUQUETTE et Brendan CHABANNES. *(Xe)L^AT_EX appliqué aux sciences humaines*. Tampere : Atramenta. 268 p. ISBN : 978-952-273-073-2.
 - [47] Gérard TISSEAU et Jacques DUMA. *Tikz pour l'impatient*. <http://math.et.info.free.fr/TikZ/index.html>. 2017.
-

4.4.5 Un ensemble d'outils

Pour bien utiliser les outils bibliographiques, il faut en comprendre un minimum le fonctionnement. La gestion des bibliographies s'articule autour d'un package `biblatex`, et d'un programme. Il existe deux programmes dans le monde de L^AT_EX, `bibtex` ou `biber`. Nous présenterons ici l'utilisation de `biber`, même si, à quelques détails près, l'utilisation de `bibtex` est similaire.

Considérons donc un fichier \LaTeX `monfichier.tex` dans lequel on veut intégrer une bibliographie. Pour ce faire, nous avons vu que les entrées bibliographiques sont stockées avec un certain format³ dans un fichier `.bib`, que nous appelons `mabibliographie.bib`.

Notre fichier \LaTeX ressemble à quelque chose comme :

```
\documentclass[french]{article}
...
\usepackage[backend=biber]{biblatex}
\bibliography{mabibliographie.bib}

...
\begin{document}

% textes où on effectue des citations
...

\printbibliography

\end{document}
```

Pour obtenir le PDF de sortie que l'on souhaite, alors il faut procéder comme suit :

- Compilation `pdflatex` du document `monfichier.tex`

```
user $> pdflatex monfichier.tex
```

- Compilation `biber` du document `monfichier`

```
user $> biber monfichier
```

Attention, il est important ici de ne pas mettre l'extension `.tex`.

- Deux compilations `pdflatex` du document `monfichier.tex`.

3. Le format dépend, à la marge, du logiciel `biber` ou `bibtex`.

Remarque 4.1 :

On notera que tous les programmes sont lancés sur le fichier `.tex` (avec ou sans l'extension) et jamais sur le fichier `mabibliographie.bib`.

4.5 INDEX

\LaTeX fournit aussi des mécanismes de construction d'index.

4.5.1 Index simple

Pour permettre de générer un index, il vous faudra faire appel au package `makeidx` [46] et ajouter dans le préambule la commande `\makeindex`. Ensuite, on ajoutera la commande `\printindex` à l'endroit où on souhaite voir apparaître l'index (c'est généralement fait à la fin du document, après la bibliographie).

4.5.2 La commande `\index`

Lorsque l'on désire ajouter un mot à un index, il suffit alors d'ajouter la commande `\index` dans le texte pour générer l'entrée dans l'index :

```
\index{<mot>}
```

Ainsi un document permettant la génération d'un index doit ressembler à ce qui suit :

```
\documentclass[french]{article}
\usepackage{makeidx}
\usepackage{babel}

\makeindex
\begin{document}
Un mot\index{mot} dans mon texte que je souhaite ajouter à
'lindex du document.
Générer un index\index{index} est très simple avec  $\LaTeX$ {}.

\printindex
\end{document}
```

Aucun contenu à l'endroit de l'utilisation de `\index` n'est ajouté lors de la compilation. De plus, on peut tout à fait utiliser une expression entière comme entrée de l'index, comportant des espaces. Ainsi, il sera possible d'écrire `\index{théorème de Lax}`.

On pourra aussi générer des *sous-entrée* d'une entrée de l'index en utilisant le séparateur `!`. Pour exemple, pour inscrire l'entrée *sous-entrée* en sous-entrée de *index*, il suffira d'écrire :

```
\index{index!sous-entrée}
```

Enfin, si on souhaite afficher quelque chose de légèrement différent du texte utilisé pour le trie de l'index, on utilisera le caractère `@`. Par exemple, si on souhaite indexer le mot \LaTeX tout en affichant le logo, on pourra utiliser le code suivant :

```
\index{LaTeX@LaTeX}
```

Ce mécanisme peut-être utile aussi pour indexer des symboles ou des lettres d'autres alphabets :

```
Je place la lettre  $\beta$  \index{beta@ $\beta$ } et je l'indexe.  
Ce caractère sera  
trié et mis à la place de \emph{beta} dans l'index.
```

4.5.3 Formatage de l'index

Pour formater l'index, comme pour générer la bibliographie (voir section 4.4), on utilisera un logiciel externe. Même s'il existe le logiciel `makeindex`, on lui préférera le programme `xindy` [20]⁴ bien plus complet, notamment pour la gestion des diverses langues et le codage UTF-8.

Pour la compilation, il faudra alors procéder en plusieurs étapes :

1. tout d'abord il faudra compiler le fichier `.tex` avec son moteur favori :

```
user $ pdflatex document.tex
```

Ceci va créer un fichier `.idx` qui contient l'ensemble des entrées de l'index.

4. D'ailleurs `xindy` est un programme général, pas seulement utilisable avec \LaTeX .

2. Il faudra ensuite exécuter `texindy` (version de `xindy` adaptée à son utilisation avec \LaTeX) une fois sur le fichier `.idx` :

```
user $> texindy -L french document.idx
```

Cette exécution va générer un troisième fichier `document.ind`.

3. Enfin, il faudra recompiler le document avec votre moteur favori :

```
user $> pdflatex document.tex
```

Tout cela peut évidemment s'automatiser avec les logiciels comme \TeX Studio (voir section 2.3.1) ou bien des programmes comme `latexmk` (voir section 3.2.1).

L'apparence des index est configurable, et on peut en créer plusieurs par document (avec le package `imakeidx` [3] par exemple), mais cela dépasse le cadre de cette introduction à \LaTeX .

LES MATHÉMATIQUES

Un des nombreux avantages à l'utilisation de \LaTeX est la composition des formules mathématiques. En effet, son inventeur D. E. K NUTH , étant lui-même mathématicien, il a développé un langage bien adapté aux mathématiques.

Ici, pour rester le plus simple et concis possible, nous supposons que nous chargeons les packages `mathtools`[16] (extension du package `amsmath`[44]), et `amssymb` [`ctan-amssymb`]. En effet, ces packages ajoutent aux possibilités \LaTeX de base des fonctionnalités très utiles.

```
\usepackage{mathtools,amssymb}
```

préambule

5.1 LES MODES MATHÉMATIQUES

La composition des mathématiques est radicalement différente de celle des mots. C'est pour cela qu'il faut indiquer au moteur de composition, c'est-à-dire à `pdf \LaTeX` , que l'on souhaite composer des mathématiques. Pour cela, \LaTeX dispose de deux modes mathématiques :

- l'un est utilisé pour composer les formules mathématiques dans le corps du texte, c'est le mode `math`, et on y rentre et on y sort grâce au caractère `$`. Ainsi, on écrit nos commandes pour écrire nos formules en mode `math` entre deux `$... $` comme montré ci après.¹

1. En réalité les dollars sont un raccourcis pour l'environnement `\begin{math} ... \end{math}`

On compose des maths dans le corps du texte grâce aux dollars

`$f(x)=x^2$`.

On compose des maths dans le corps du texte grâce aux dollars $f(x) = x^2$.

- L'autre mode est utilisé pour composer les formules *hors texte* seules sur une ligne. C'est le mode `displaymath` et on y rentre avec le mécanisme : `\[... \]` comme montré ci après.²

Le même exemple que précédemment mais en mode `\lstinline+displaymath+ \[f(x)=x^{2}\]`.

Le même exemple que précédemment mais en mode `displaymath`

$$f(x) = x^2$$

5.1.1 Fonctionnement singulier

En mode mathématique, le fonctionnement est très particulier. Les espaces entre les lettres ne sont plus les mêmes, les lettres elles-mêmes sont composées dans un style différent (pour en faciliter la lecture), et surtout les espaces sont *complètement ignorés*. Pour les composer lorsque cela est nécessaire, il faudra utiliser des commandes d'espacement que nous présentons en section 5.2.2.

Voici un exemple montrant que les espaces sont ignorés.

```
\[
f x t e
\]
```

f x t e

}.

2. Là encore, il s'agit d'un raccourci à l'environnement `\begin{displaymath} ... \end{displaymath}`.

5.2 LES BASES DE LA COMPOSITION DE MATHÉMATIQUES

Nous présentons ici quelques mécanismes fondamentaux de la composition de mathématiques.

5.2.1 Indices et exposants

Dès que l'on souhaite écrire des mathématiques, il nous faut savoir comment mettre des indices et des exposants. Cela se fait très simplement avec \LaTeX

Pour les indices c'est le caractère spécial `_` qui sert :

```
\[
x_i
\]
```

$$x_i$$

Pour bien être à l'aise avec les indices, il faut bien comprendre les *groupes*. En \LaTeX , les groupes peuvent être délimités par les accolades. Le caractère spécial `_` prend deux arguments : le groupe avant et le groupe après :

\langle *groupe à indiquer* \rangle_{\langle *groupe en indice* $\rangle}$

De la même façon, les exposants se composent avec le caractère spécial `^`.

\langle *groupe* \rangle^{\langle *groupe en exposant* $\rangle}$

Si on n'utilise pas d'accolades de groupement, alors ce sont les premiers caractères qui sont mis en indice.

```
 $\$x^2+y^2=1\$, \$x_1=x_2\$, \$x_1^2+x_2^2=1\$\$ 
```

$$x^2 + y^2 = 1, x_1 = x_2, x_1^2 + x_2^2 = 1$$

Pour comprendre l'utilité de groupe, rien de mieux que quelques exemples.

```
\begin{itemize}
\item  $a_{ij}$ 
\item  $x^{y^z}$ 
\item différence entre  $(x+y)^n$  et  $\{(x+y)\}^n$ 
\end{itemize}
```

-
- a_{ij}
 - x^y^z
 - différence entre $(x + y)^n$ et $(x + y)^n$
-

5.2.2 Les espaces en mode mathématique

Il est possible de modifier l'espacement en mode mathématique afin d'éloigner ou de rapprocher certains symboles. Nous présentons les commandes d'espacement dans le tableau 5.1.

<code>\,</code>	petit espace	<code>\!</code>	petit espace <i>négatif</i>
<code>\:</code>	espace moyen	<code>\quad</code>	petit espace interformules
<code>\;</code>	grand espace	<code>\qquad</code>	espace moyen interformules

TABLE 5.1 – Commandes d'espacement en mode mathématique.

L'exemple suivant permet de comprendre à quoi peuvent servir les plus courantes de ces quelques commandes.

```
\newcommand\ Dx{\, \mathrm{d}x}
Montrer que :
\[\forall x \in \mathbb{R}, \quad \int_0^x f(x) \, Dx = F(x) - F(0).\]
```

Montrer que :

$$\forall x \in \mathbb{R}, \quad \int_0^x f(x) \, dx = F(x) - F(0).$$

Il vaut mieux tout de même éviter de les utiliser à tort et à travers dans notre document. Si l'utilisation d'une de ces commandes est très récurrente et structurale, alors il vaudra mieux passer par le mécanisme de définition de nouvelle commande.

5.2.3 Du texte à l'intérieur d'une formule

Surtout en mode `displaymath`, il est souvent utile de mettre du texte. Ceci peut se faire avec la commande `\text{ ... }`.

```
\[
f_{[x_{i}, x_{i+1}]} \text{ est croissante pour tout } i \in \{1, \dots, N\}
\]
```

$f_{[x_i, x_{i+1}]}$ est croissante pour tout $i \in \{1, \dots, N\}$

Texte comme indice ou exposant — L'objet composé par `\text{ ... }` peut aussi servir à mettre en indice ou en exposant.

`\[\sum F_{\text{ext}}=\vec{a}\]`

$$\sum F_{\text{ext}} = \vec{a}$$

Math dans du texte dans des maths... — Comme la commande `\text` repasse en mode texte localement, pour écrire des maths dedans, il faut se remettre en mode math à l'intérieur de la commande.

`\[`
`\partial_{s} f(x) = \frac{\partial}{\partial x_0} f(x) \quad \text{pour } x = x_0 + I x_1.`
`\]`

$$\partial_s f(x) = \frac{\partial}{\partial x_0} f(x) \quad \text{pour } x = x_0 + I x_1.$$

5.2.4 Lettres grecques

Pour la composition des mathématiques, il est nécessaire d'avoir accès aux lettres grecques, et cela même sans utiliser un alphabet grec. Tout cela est prévu par \LaTeX , et les lettres grecques sont accessibles via des commandes. Celles-ci sont présentées dans le tableau 5.2.

Remarque 5.1 :

Attention, ces commandes ne sont disponibles qu'en mode mathématique.

5.2.5 Symboles d'opérateurs binaires

Il est souvent nécessaire d'utiliser des opérateurs binaires autres que les simples `+` ou `-` directement accessibles via les touches du clavier. Les commandes pré-

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

Majuscules							
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 5.2 – Lettres grecques

sentées dans le tableau 5.3 permettent d’obtenir tout un ensemble de symboles d’opérateurs binaires.

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangle	<code>\bigtriangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	∇	<code>\bigtriangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\triangleleft	<code>\lhd</code> ^a	\circ	<code>\bigcirc</code>
\circ	<code>\circ</code>	\wedge	<code>\wedge</code>	\triangleright	<code>\rhd</code> ^a	\dagger	<code>\dagger</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>	\triangleleft	<code>\unlhd</code> ^a	\ddagger	<code>\ddagger</code>
\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\triangleright	<code>\unrhd</code> ^a	\amalg	<code>\amalg</code>

^a Requiert l’extension `latexsym`.

TABLE 5.3 – Symboles d’opérateurs binaires

5.2.6 Des symboles, lettres et opérateurs utiles

Dans cette section, nous donnons à titre indicatif des ensembles de symboles ou de construction de symboles mathématiques, de lettres, et d’opérateurs qui sont utiles à tou·te·s mathématicien·ne·s.

Symboles de relations binaires

Les commandes présentées dans le tableau 5.4 permettent d'obtenir des symboles de relations binaires tels que les relations d'ordres, les relations d'inclusion, etc.

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
$<$	<code>\prec</code>	$>$	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\leq	<code>\preceq</code>	\geq	<code>\succeq</code>	\approx	<code>\simeq</code>	$ $	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>		<code>\Join</code> ^a
\sqsubset	<code>\sqsubset</code> ^a	\sqsupset	<code>\sqsupset</code> ^a	\neq	<code>\neq</code>	\smile	<code>\smile</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\doteq	<code>\doteq</code>	\frown	<code>\frown</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>		
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>				

^a Requier l'extension `latexsym`.

TABLE 5.4 – Symboles de relations binaires

On peut obtenir la négation de ces symboles grâce à la commande `\not` :

```
\[x\notin F\]
```

$$x \notin F$$

Flèches

Les flèches sont extrêmement utiles en mathématique. \LaTeX en propose un large éventail avec les commandes qui sont présentées dans le tableau 5.5.

Symboles divers

Les commandes présentées dans le tableau 5.6 permettent d'obtenir des symboles divers souvent utiles à la composition de mathématiques.

Accents

Dans cette section nous présentons les commandes qui permettent de placer au dessus ou en dessous de lettre ou groupe de symboles, des éléments. C'est le cas en particulier des accents qui nous listons dans le tableau 5.7. À noter que ces accents s'obtiennent en mode mathématique.

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\Uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\Downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\Updownarrow	<code>\updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto	<code>\leadsto</code> ^a		

^a Requierit l'extension `latexsym`.

TABLE 5.5 – Flèches

\aleph	<code>\aleph</code>	$'$	<code>\prime</code>	\forall	<code>\forall</code>	∞	<code>\infty</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\square	<code>\Box</code> ^a
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\diamond	<code>\Diamond</code> ^a
\jmath	<code>\jmath</code>	$\sqrt{\quad}$	<code>\surd</code>	\flat	<code>\flat</code>	\triangle	<code>\triangle</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\clubsuit	<code>\clubsuit</code>
\wp	<code>\wp</code>	\perp	<code>\bot</code>	\sharp	<code>\sharp</code>	\diamondsuit	<code>\diamondsuit</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\backslash	<code>\backslash</code>	\heartsuit	<code>\heartsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	∂	<code>\partial</code>	\spadesuit	<code>\spadesuit</code>
\mho ^a	<code>\mho</code> ^a	\dots	<code>\dots</code>	\vdots	<code>\vdots</code>	\cdots	<code>\cdots</code>

^a Requierit l'extension `latexsym`.

TABLE 5.6 – Symboles divers

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

TABLE 5.7 – Accents

Les lettres accentuées sont très utilisées en mathématiques pour augmenter le nombre de symbole, pour déterminer une transformation, etc.

Notons que les caractères i et j ont un statut singulier du fait du point qu'ils comportent. En effet, l'accent devra alors remplacer le point. Pour ce faire, \LaTeX fournit les commandes `\imath` et `\jmath` qui sont les lettres sans leurs points respectifs.

Si on veut accentuer avec le chapeau ou le tilde des mots alors il existe les deux commandes `\widehat` et `\widetilde` qui s'utilisent comme ceci :

```
\[\widehat{xyz}\quad \widetilde{xyz}\]
```

$$\widehat{xyz} \quad \widetilde{xyz}$$

Toujours dans les mécanismes d'ajout de symboles sur des lettres, on dispose aussi des commandes pour surmonter leurs arguments d'une longue flèche : `\overrightarrow` et `\overleftarrow`.

```
\[\overrightarrow{AB}\quad \overleftarrow{BA}\]
```

$$\overrightarrow{AB} \quad \overleftarrow{BA}$$

Les commandes `\overline` et `\underline` permettent de surligner et de souligner :

```
$ \overline{z+1} = \bar{z} + 1 $,  
$ \underline{\alpha + \beta} $
```

$$\overline{z+1} = \bar{z} + 1, \underline{\alpha + \beta}$$

Enfin, les commandes `\overbrace` et `\underbrace` placent des accolades sur ou sous le texte. On peut ajouter un commentaire en mettant un exposant après la commande `\overbrace` ou un indice après la commande `\underbrace` :

```
\[a^n = \overbrace{a \times a \times \cdots \times a}^{\text{\mbox{\$n\$ fois}}}\]
```

$$a^n = \overbrace{a \times a \times \dots \times a}^{n \text{ fois}}$$

5.2.7 Superposer deux symboles

Il est possible de superposer deux symboles grâce à la commande :

`\stackrel{\langle dessus \rangle}{\langle dessous \rangle}`

`$ x \stackrel{f}{\longmapsto} f(x) $`

$$x \stackrel{f}{\longmapsto} f(x)$$

5.2.8 Modification de style

Les ensembles

On peut composer les symboles utilisés pour les ensembles usuels. Historiquement, ces ensembles étaient composés dans les textes imprimés par des lettres grasses. Ceci s'obtient avec la commande `\mathbf` :

`\[`
`\mathbf{N}, \quad \mathbf{R}, \quad \mathbf{C}`
`\]`

N, R, C

Celle-ci peut évidemment s'utiliser avec les lettres minuscules.

Avec le développement de l'enseignement, il a fallu retranscrire ces notations sur un tableau (noir à craie pour l'époque). Ainsi, il est désormais de coutume de composer les ensembles dans une fonte particulière. En \LaTeX , ceci s'obtient grâce à la commande `\mathbb{...}` comme *blackboard* :

`\[`
`\mathbb{N}, \quad \mathbb{R}, \quad \mathbb{C}`
`\]`

N, R, C

Les divers styles

Il existe des commandes analogues aux changements de styles dans le texte mais pour le mode mathématiques (voir le tableau 5.8).

<code>\$ x + y + 2^{n}\cos x\$</code>	$x + y + 2^n \cos x$
<code>\$\$\mathit{x + y + 2^{n}\cos x}\$\$</code>	$x + y + 2^n \cos x$
<code>\$\$\mathrm{x + y + 2^{n}\cos x}\$\$</code>	$x + y + 2^n \cos x$
<code>\$\$\mathbf{x + y + 2^{n}\cos x}\$\$</code>	$\mathbf{x + y + 2^n \cos x}$
<code>\$\$\mathsf{x + y + 2^{n}\cos x}\$\$</code>	$x + y + 2^n \cos x$
<code>\$\$\mathhtt{x + y + 2^{n}\cos x}\$\$</code>	$x + y + 2^n \cos x$

TABLE 5.8 – Commandes de changement de style

Comme vous pouvez le constater, les commandes de changement de style n’agissent que sur les lettres, les chiffres et les lettres grecques majuscules.

Enfin, il est souvent utile de pouvoir composer les lettre majuscules dans un style calligraphié. Ceci se fait avec la commande `\mathcal` :

```
\[
  \mathcal{N}, \quad \mathcal{R}, \quad \mathcal{C}
\]
\[\mathcal{ABCDEFGHIJKLMNPOQRSTUVWXYZ}\]
```

N, R, C

ABCDEFGHIJKLMNPOQRSTUVWXYZ

Remarque 5.2 :

Il est d’usage de composer les constantes mathématiques dans le style roman. Ainsi on écrira :

```
\[
  \mathrm{e}^{\mathrm{i}\pi}+1=0
\]
```

$e^{i\pi} + 1 = 0$

5.2.9 Les fractions

Les fractions se composent avec la commande :

`\frac{⟨numérateur⟩}{⟨dénominateur⟩}`

```
\[
\frac{1+x^2}{1-x^2}=?
\]
```

$$\frac{1+x^2}{1-x^2} = ?$$

5.2.10 Les racines

Les racines se composent avec la commande :

`\sqrt[⟨ordre⟩]{⟨argument⟩}`

```
\[
\sqrt{x^2+y^2}, \quad \sqrt[n]{a_i}
\]
```

$$\sqrt{x^2 + y^2}, \quad \sqrt[n]{a_i}$$

5.2.11 Délimiteurs

Un des mécanismes de \LaTeX très utile est celui des délimiteurs. En effet, en mathématiques, on groupe des parties d'équations par des parenthèses, des crochets, des accolades, etc. Ces éléments doivent être de taille adaptée au groupe qu'ils contiennent.

En avec \LaTeX , on compose ce type d'éléments avec les délimiteurs. Les délimiteurs doivent toujours aller par paire : un délimiteur ouvrant et un délimiteur fermant. Un délimiteur ouvrant s'obtient par la commande `\left` suivie du délimiteur, et un délimiteur fermant par la commande `\right` suivie du délimiteur. Bien entendu, on peut imbriquer les paires de délimiteurs si nécessaire.

Les éléments qui peuvent se mettre derrière `\left` et `\right` sont listés dans le tableau 5.9.

Illustrons ce mécanisme par des exemples.

(())	↑	<code>\uparrow</code>
[[]]	↓	<code>\downarrow</code>
{	<code>\{</code>	}	<code>\}</code>	↕	<code>\updownarrow</code>
⌊	<code>\lfloor</code>	⌋	<code>\rfloor</code>	⇕	<code>\Uparrow</code>
⌈	<code>\lceil</code>	⌉	<code>\rceil</code>	⇓	<code>\Downarrow</code>
⟨	<code>\langle</code>	⟩	<code>\rangle</code>	↕	<code>\Updownarrow</code>
/	/	\	<code>\backslash</code>	∥	<code>\ </code>

TABLE 5.9 – Délimiteurs

```
\left(\frac{1}{1+x}\right)
\left(\frac{1}{1+x}\right)
```

$$\left(\frac{1}{1+x}\right)$$

$$\left[\frac{1}{1+x}\right]$$

Si on ne veut rien mettre d'un côté ou de l'autre, alors il faut utiliser le délimiteur spécial : ..

```
\[
f(x) =
\left\{
\begin{array}{rl}
x & \text{si } x \geq 0 \\
-x & \text{si } x < 0
\end{array}
\right.
\right.
```

$$f(x) = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$$

5.2.12 Symboles de taille variable

Les symboles présentés dans le tableau 5.10 sont de *taille variable* : il n'ont pas la même taille dans l'environnement `math` et dans l'environnement `displaymath`.

Ces symboles sont plus petits en mode `math` et plus gros en mode `displaymath`

$\Sigma \Sigma$	<code>\sum</code>	$\cap \cap$	<code>\bigcap</code>	$\odot \odot$	<code>\bigodot</code>
$\Pi \Pi$	<code>\prod</code>	$\cup \cup$	<code>\bigcup</code>	$\otimes \otimes$	<code>\bigotimes</code>
$\amalg \amalg$	<code>\coprod</code>	$\sqcup \sqcup$	<code>\bigsqcup</code>	$\oplus \oplus$	<code>\bigoplus</code>
$\int \int$	<code>\int</code>	$\vee \vee$	<code>\bigvee</code>	$\bigcup \bigcup$	<code>\bigcupplus</code>
$\oint \oint$	<code>\oint</code>	$\wedge \wedge$	<code>\bigwedge</code>		

TABLE 5.10 – Symboles de taille variable

pour que dans le corps du texte, cela ne perturbe pas l'interlignage. On note aussi que pour ces symboles, les indices et exposant (qui en fait sont leur bornes) ne sont pas disposés de la même façon dans les deux modes.

```

 $\sum_{i=1}^{+\infty} x_i \quad \int_0^{+\infty} f(x) dx; \mathrm{d}$ 
 $\sum_{i=1}^{+\infty} x_i \quad \int_0^{+\infty} f(x) dx; \mathrm{d}$ 

```

$$\sum_{i=1}^{+\infty} x_i \quad \int_0^{+\infty} f(x) dx$$

$$\sum_{i=1}^{+\infty} x_i \quad \int_0^{+\infty} f(x) dx$$

5.2.13 Fonctions et opérateurs

Il y a des conventions particulières pour les opérateurs ou les fonctions en mathématiques. Pour cela, \LaTeX dispose de la notion d'opérateurs qui permettent de produire ce genre de chose :

```

 $\lim_{x \rightarrow +\infty} \ln x = +\infty$ 

```

$$\lim_{x \rightarrow +\infty} \ln x = +\infty$$

On a plusieurs opérateurs prédéfinis que nous listons dans le tableau 5.11.

Certains de ces opérateurs se comportent comme les symboles à taille variable. Ainsi, la composition de l'expression précédente donnera mode `math` simple

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

TABLE 5.11 – Fonctions et opérateurs

```
$
\lim_{x\to+\infty}\ln x = +\infty
$
```

```
limx→+∞ ln x = +∞
```

Déclaration de nouveaux opérateurs

Pour créer de nouveaux opérateurs, le package `mathtools` fournit deux commandes :

```
\DeclareMathOperator{\<nom de la commande>}{<text à afficher>}
\DeclareMathOperator*{\<nom de la commande>}{<text à afficher>}
```

La version *non étoilée* permet de construire des opérateurs comme les fonctions du type de `\sin`. La version étoilée, elle, permet de faire des opérateurs comme `\lim` avec les exposants et indices respectivement au dessus et en dessous de l'opérateur.

```
\DeclareMathOperator{\Reel}{Re}
\DeclareMathOperator*{\Tot}{Toto}
```

```
\[\Reel z \]
\[\Tot_{x= 3}^N\]
```

Re z
N
Toto
x=3

5.2.14 Les matrices

Le package `mathtools` permet de composer facilement les matrices. Nous disposons de plusieurs environnements pour le faire.

```
\[
{\mathcal A} =
\begin{matrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{matrix}
\]
```

$$\mathcal{A} = \begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix}$$

Mais on peut avoir des délimiteurs autour du tableau :

```
\[
{\mathcal A} =
\begin{pmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{pmatrix}
=
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix}
\]
```

$$\mathcal{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

5.3 LES ÉQUATIONS NUMÉROTÉES ET LES RÉFÉRENCES

Il est nécessaire de pouvoir faire référence aux équations le long du texte. Pour cela, les mécanismes de références croisées vus en section 4.1 fonctionnent aussi. Cependant, il ne faut plus utiliser le mode `displaymath` simple avec les

crochets ouvrant et fermant `\[... \]`, mais il faut utiliser l'environnement :

```
\begin{equation}
...
\end{equation}
```

Cette environnement va numéroter l'équation et permet d'y ajouter un *label* pour y faire référence. Cependant, puisque la numérotation des équations par l'environnement `equation` se fait entre parenthèse, on préférera utiliser `\eqref` au lieu de `\ref` lorsqu'on y fera référence.

Ainsi, nous pouvons composer ceci :

```
\begin{equation}\label{eq:belle}
\mathrm{e}^{\mathrm{i}\pi}+1=0
\end{equation}
```

Et faire référence à cette belle équation~`\eqref{eq:belle}`.

$$e^{i\pi} + 1 = 0 \tag{5.1}$$

Et faire référence à cette belle équation (5.1).

5.3.1 Les équations alignées

Lors des démonstrations, il est souvent nécessaire d'aligner un ensemble d'équation par rapport à un symbole (*i.e.* le signe égal). Le package `mathtools` fournit un environnement qui permet d'aligner les équations et de les numéroter.

Un exemple parlera bien mieux qu'une longue explication :

```
\begin{align}
\int_1^2 x^2\; \mathrm{d} x
&= \left[ \frac{x^3}{3} \right]_1^2 \label{eq:
premiereetape} \\
&= \frac{2^3}{3} - \frac{1^3}{3} \\
&\quad \nonumber \\
&= \frac{8}{3} - \frac{1}{3} \\
&\quad \nonumber \\
&= \frac{7}{3} \label{eq:fin}
\end{align}
```

On passe de~`\eqref{eq:premiereetape}` à~`\eqref{eq:fin}` par simple calcul.

$$\int_1^2 x^2 dx = \left[\frac{x^3}{3} \right]_1^2 \quad (5.2)$$

$$= \frac{2^3}{3} - \frac{1^3}{3}$$

$$= \frac{8}{3} - \frac{1}{3}$$

$$= \frac{7}{3} \quad (5.3)$$

On passe de (5.2) à (5.3) par simple calcul.

Si on souhaite simplement aligner un ensemble d'équation, alors il suffit d'utiliser la *version étoilée* de l'environnement :

```
\begin{align*}
\ln xy &= \ln x + \ln y \\
\exp(x+y) &= \exp(x) \exp(y)
\end{align*}
```

$$\ln xy = \ln x + \ln y$$

$$\exp(x + y) = \exp(x) \exp(y)$$

5.4 THÉORÈMES, DÉFINITIONS, ETC.

Pour composer des théorèmes, des définitions, des propositions, etc. auxquels on peut faire référence et qui ont une typographie qui les différencie du reste du texte, \LaTeX fournit un mécanisme qui permet de définir des environnements.

```
\newtheorem{<nom de l'environnement>}{<Texte à imprimer>}
```

Par exemple, on peut définir l'environnement `theoreme` :

```
\newtheorem{Theoreme}{Théorème}

\begin{Theoreme}[Titre du théorème]\label{thm:test}
En voilà un beau théorème !
\end{Theoreme}
```


Et nous pouvons même faire référence au théorème~\ref{thm:test}.

Théorème 1 (Titre du théorème). *En voilà un beau théorème!*

Et nous pouvons même faire référence au théorème 1.

L'environnement créé par la commande `\newtheorem` réalise les actions suivantes :

1. il imprime *⟨Texte à imprimer⟩* (deuxième argument de la commande `\newtheorem`);
2. puis un numéro (à chaque environnement créé est associé un compteur dédié qui est incrémenté à chaque utilisation de l'environnement);
3. puis, éventuellement, son argument optionnel entre parenthèses;
4. enfin, il imprime son contenu, préalablement mis en italique.

On peut ainsi créer autant d'environnements que nécessaire, un pour les théorèmes, un pour les définitions, un pour les lemmes, un pour les hypothèses, etc., chacun ayant son propre compteur.

Évidemment, l'apparence des théorèmes est entièrement personnalisable, mais cela dépasse le cadre de cette trop courte introduction.

SUPPORTS DE PRÉSENTATION AVEC BEAMER

Beamer[32] est une classe \LaTeX spécialement conçue pour créer des présentations destinées à être vidéo-projetées. Ce package est bien utile et souvent plus pratique lorsque l'on fait des exposés mathématiques. En effet :

- tous les outils classiques de \LaTeX présentés dans ce document, sont disponibles notamment pour la composition de mathématiques, la gestion de bibliographie, la création de sommaire, la structuration du document, etc.
- Les présentations produites avec beamer sont portables : on produit un PDF lisibles de la même façon quelque soit l'ordinateur utilisé.
- Ce package permet de produire des présentations vivantes avec quelques animations (évidemment on a du PDF, cela reste sommaire par rapport à des logiciels dédiés aux présentations animées).
- Il y a énormément de thèmes prédéfinis, et on trouve très facilement chaussure à son pied. Tout est largement personnalisable.
- On peut très simplement gérer les différentes versions d'une présentation.
- La documentation [32], bien qu'en anglais, est très complète, avec des conseils, et des exemples très utiles.

Dans ce chapitre, nous mettrons simplement le pied à l'étrier, et il faudra nécessairement compléter cela par les innombrables exemples présents dans la documentation ou sur le web. Cette présentation de beamer est très inspiré du site :

<https://open-freax.fr/initiation-beamer/>

6.1 PRÉPARATION DU DOCUMENT : LE PRÉAMBULE

Ici il faudra charger la classe beamer et choisir un thème.

```
\documentclass[ options ]{beamer}
\usetheme{theme} % on choisit un thème à utiliser (non
obligatoire)
\usecolortheme{couleur} % si besoin un autre jeu de couleurs (
non obligatoire)
\usefonttheme{fonte} % ainsi que d'autres fontes (non
obligatoire)
% suite du préambule classique de LaTeX
```

préambule

Pour choisir le thème, on pourra aller voir la documentation [32, p. 148] ou bien le site suivant :

<https://hartwork.org/beamer-theme-matrix/>

Il faudra aussi aller voir la documentation pour les jeux de couleurs et les fontes disponibles.

6.1.1 Les informations de son documents

La diapositive de début de la présentation est générée avec les informations qu'on indique dans le préambule. Plutôt qu'un long discours, nous présentons ici les quelques champs à remplir.

```
\title[titre court]{titre}
\subtitle[sous-titre court]{sous-titre}
\author[auteur(s) court(s)]{auteur(s)/trice(s)}
\institute[institut court]{institut}
\date[date courte]{date}
\titlegraphic{\includegraphics{fichier}}
\subject{sujet}
\keywords{mot(s) clé(s)}
```

préambule

Il y a quelques règles à suivre : s'il y a plusieurs auteurs ou autrices, il faut les séparer par `\and`. Les champs `\subject` et `\keywords` permettent de remplir les propriétés du document PDF qui sera généré. Cela n'apparaîtra pas dans les diapositives.

6.2 LES DIAPOSITIVES

Une présentation vidéo-projetée est nécessairement un enchaînement de diapositives (*slides*). La syntaxe est ici un environnement `frame`.

Remarque 6.1 :

Pour structurer la présentation, on utilisera comme dans un document \LaTeX classique les `\section`, `\subsection`, etc. Cependant, il faudra veiller à ne pas les faire apparaître à l'intérieur des environnements `frame`.

Le titre de la diapositives se définit avec la commande `\frametitle` et `\framesubtitle`.

Voici un exemple d'utilisation de tout cela.

```

\section{Ma section}
\subsection{Une sous-section}

\begin{frame}
  \frametitle{\insertsection} % qui va imprimer Ma section
  \framesubtitle{\insertsubsection} % qui va imprimer Une sous
    section
  Du contenu
\end{frame}

\begin{frame}
  \frametitle{Un titre complètement différent}
  \framesubtitle{Un sous titre particulier}
  Du contenu
\end{frame}

```

On pourra voir un exemple de diapositive avec le thème `Rochester` et le thème de couleur `seagull` en figure 6.1.

La diapositive est découpé en zones :

- un en-tête et un pied de page ;
- une marge gauche et une marge droite ;
- des barres de navigation ;
- des symboles de navigation ;
- un emplacement pour un logo ;
- un titre de cadre, ainsi qu'un sous-titre ;

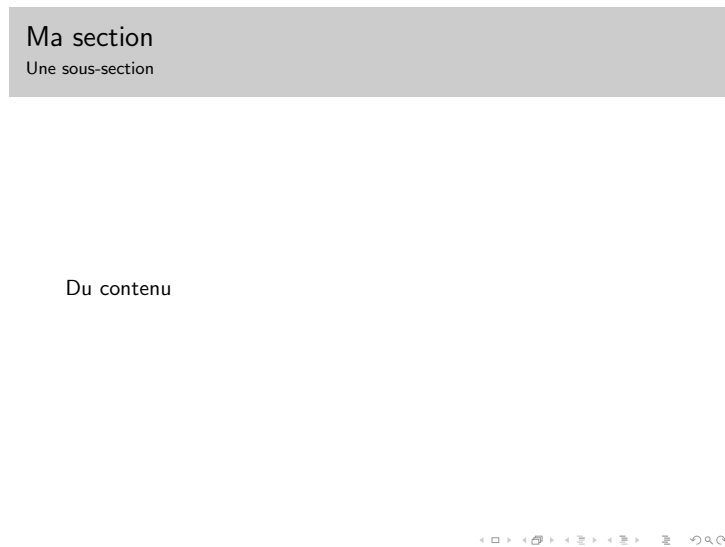


FIGURE 6.1 – Exemple de diapositive avec beamer avec le thème Rochester et le thème de couleur seagull

- un fond;
- une zone de contenu.

On peut avec un peu de travail de lecture de la documentation, personnaliser toutes ces zones, même si les paramètres par défaut sont souvent suffisant.

Remarque 6.2 :

Par défaut, des symboles de navigation sont apparents. Or, il ne servent pas à grand chose la plupart du temps. Il est à notre sens, préférable de les masquer :

```
\setbeamertemplate{navigation symbols}{}
```

6.2.1 Diapositive de titre

Pour générer la première diapositive, la page de titre, il suffit d'utiliser la commande

```
\titlepage
```

à l'intérieur d'un environnement frame.

```
\begin{frame}
  \titlepage
\end{frame}
```

On peut voir un exemple généré avec le thème Rochester et le thème de couleur seagull en figure 6.2.



FIGURE 6.2 – Page de titre avec beamer avec le thème Rochester et le thème de couleur seagull

6.3 TABLE DES MATIÈRES

Un autre élément très utile de \LaTeX est la génération automatique des tables des matières. Ce mécanisme est décliné aussi avec la classe beamer. Ici, l'intérêt est qu'on peut utiliser plusieurs fois : au début pour annoncer le plan, entre deux sections pour montrer l'évolution et là où on en est, etc.

La commande a été redéfinie pour permettre l'ajout d'options.

```
\tableofcontents[\langle options \rangle]
```

Les options sont alors :

- `currentsection` (et `currentsubsection`) qui permet d'afficher la section (ou sous-section) en cours, en estompant légèrement les autres morceaux de la table des matières;

- `hideothersubsections` qui, comme son nom l'indique, masque les autres sous-sections autres que celle en cours ;
- `hideallsubsections` qui, lui, masque toutes les sous-sections.

Le code suivant

```
\begin{frame}
\frametitle{Table des matières}
\tableofcontents
\end{frame}
```

produit le résultat montré en figure 6.3, toujours avec le thème Rochester et le thème de couleur seagull.

Table des matières

- 1 Ma section
 - Une sous-section

- 2 Ma deuxième section
 - Une subsection
 - Une autre...

FIGURE 6.3 – La table des matières avec le thème Rochester et le thème de couleur seagull.

6.4 LES BLOCS

Beamer fournit des environnements bien utiles pour la composition des diapositives. Il s'agit des blocs, et il y en a trois types :

- le type `simpleblock` ;
- le type `alertblock` ;
- et le type `exampleblock`.

La syntaxe de ces blocs, ici pour l'environnement `block`, est la suivante :

```
\begin{block}{Titre}
  Du contenu, ce qu'on veut, du texte, des maths, des images\
  dots
\end{block}
```

Suivant le thème utilisé pour la classe `beamer` l'apparence de ces boîtes sera différentes. En figure 6.4, on peut voir l'apparence de ces blocs avec avec le thème `Rochester` et le thème de couleur `seagull`.

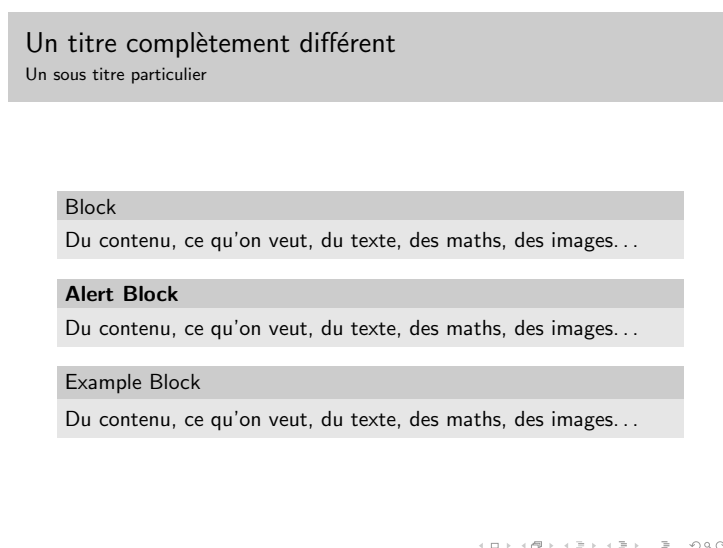


FIGURE 6.4 – Les différents blocs avec le thème `Rochester` et le thème de couleur `seagull`.

6.5 LES COUCHES

Pour rendre plus dynamiques la projection des diapositives, `beamer` introduit le concept de couche (*overlays* en anglais). Ces couches permettent de faire apparaître les éléments constituant la frame petit à petit. À la compilation, \LaTeX générera une page PDF par couche.

L'exemple le plus simple de l'utilisation des couches est l'affichage des éléments d'une liste (`itemize`, `enumerate` ou `description`) au fur et à mesure. Pour réaliser cela, il suffit de mettre à la fin de chaque `\item`, comme illustré dans l'exemple suivant :

```

\begin{frame}
  \begin{itemize}
    \item Premier élément \pause
    \item Deuxième élément \pause
    \item Troisième élément
  \end{itemize}
\end{frame}

```

Avec cet exemple, la frame est composée de trois couches.

Certaines commandes ou environnements comme :

- les changements de style (`\textbf`, `\textit` etc.);
- les listes comme on a vu, et notamment `\item`;
- les changements de couleur (`\textcolor`, `\alert`, etc.);
- les commandes et environnements spécifiques à la classe beamer (comme `block`, etc.);

acceptent un *argument* supplémentaire qui permet de spécifier les numéros de couches d'apparition de l'objet en question. Cette argument est indiqué entre < et >.

Imaginons que l'on souhaite mettre un mot d'une phrase en valeur mais dynamiquement, c'est-à-dire qu'on souhaite qu'il apparaisse en gras une fois qu'on a cliqué pour changer de diapositive. Alors il suffira d'écrire :

```
Il faut \textbf<2>{apprendre} \LaTeX{ }, c'est vital !
```

Le mot apprendre deviendra gras sur la deuxième couche.

La syntaxe pour indiquer les couches permet de tout faire ou presque. Voici quelques exemples :

- <1> : seule la couche 1 est concernée par la commande ;
- <1,3,4> : la commande sera *active* sur les couches 1, 3 et 4, mais pas la 2 ;
- <1-4> : les couches 1 à 4 (inclus) sont concernées par la commande ;
- <3-> : la commande s'applique à la couche 3 et toutes les couches suivantes de la frame ;
- <-2> : la commande s'applique du début de la frame jusqu'à la couche 2 (incluse) ;
- <-3,6,9-> : on combine le tout, et la commande s'applique aux couches 1 à 3, à la couche 6, et de la couche 9 à la dernière couche de la diapo.

Il existe aussi des commandes pour afficher ou cacher leur argument comme `\uncover` et `\only` qui respectivement garde ou non la place réservée lors que l'argument est caché.

Bref les possibilités sont assez impressionnantes. Cependant, tout ceci n'est qu'une introduction au mécanisme de couche, et pour bien le maîtriser et savoir à quoi il peut s'appliquer, il faudra aller chercher dans la documentation ou sur le web plus de précisions.

6.6 LA BIBLIOGRAPHIE

Là encore, on bénéficie de la puissance de gestion de bibliographie de \LaTeX , et notamment du package `biblatex` que nous avons présenté en section 4.4.

REPRÉSENTATIONS GRAPHIQUES

Lors de la composition d'un document, il est souvent nécessaire d'y inclure des images. Nous avons vu, en section 3.10 comment inclure des images sous forme d'un fichier (PDF, jpg, png,...) extérieur.

Dans les productions mathématiques ces images peuvent être, le plus souvent, des illustrations de concepts mathématiques ou des représentations graphiques d'expériences (par exemple numériques).

Il existe de très nombreux moyens informatiques de produire ces images. Cependant, le fait de les produire en dehors du monde de \LaTeX fait que ces *dessins* ont rarement la qualité du document produit, lui, avec \LaTeX . Dans ce chapitre, on donnera quelques pistes pour obtenir des dessins et des représentations graphiques de qualité. Là encore, cet aspect de la composition est gigantesque, et nous n'en effleurerons qu'une petite partie.

7.1 UTILISER LE BON OUTIL

Même s'il est possible de faire à peu près tout comme dessin avec \LaTeX , il est préférable d'utiliser l'outil le plus adapté. Les solutions que nous allons présenter dans ce chapitre concernent des dessins mathématiques, géométriques, qui se décrivent bien. Pour tout ce qui est purement graphique, nous ne serons que conseiller des outils plus adaptés au graphisme.

Il en existe de très bons, en logiciels libres et multiplateforme. Nous ne mentionnerons ici uniquement le logiciel Inkscape

<https://inkscape.org/fr/>

qui permet de faire *du dessin vectoriel* qui n'a pas de défaut de résolution et donc qui permet d'obtenir une très bonne qualité d'image.

De plus, Inkscape permet, avec un *plugin* dédié, d'insérer du code \LaTeX dans le dessin.

7.1.1 \LaTeX pour des dessins « géométriques »

Les outils que fournit \LaTeX , que nous ne présenterons pas tous ici, tant l'offre est gigantesque, sont adaptés à des représentations graphiques géométriques qui peuvent se décrire mathématiquement.

7.2 TIKZ

Nous présenterons dans ce document extrêmement rapidement le package *Tikz* [**ctan-tikz**]. *Tikz* étant un package, il s'utilise comme tout autre package, en déclarant dans la préambule du document :

```
\usepackage{tikz}
```

préambule

7.2.1 *Tikz* pour l' impatient

Dans ce document, nous nous contenterons d'effleurer l'utilisation de ce package pour en présenter l'esprit et quelques éléments de bases. Pour une introduction bien plus complète en français, nous conseillons le PDF *Tikz pour l' impatient* [47] à découvrir à l'adresse suivante :

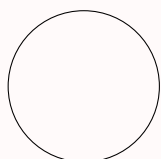
<http://math.et.info.free.fr/TikZ/index.html>.

7.2.2 Premier dessin

Pour réaliser une figure (*picture* en anglais) dans un document \LaTeX , l'environnement à utiliser est `tikzpicture`. À l'intérieur de cet environnement se trouve du code \LaTeX avec une syntaxe propre à *Tikz*.

Voici un premier exemple de tracé d'un simple cercle.

```
\begin{tikzpicture}
  \draw (0,0) circle (1);
\end{tikzpicture}
```



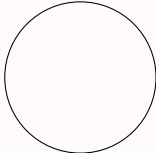
On constate que la syntaxe ne respecte pas tout à fait les conventions du \LaTeX *standard*. Dans ce document, nous ne détaillerons pas toutes les détails du langage de Tikz, nous essaierons juste, par l'exemple, de montrer les potentialités et l'utilité de ce package.

Placement de la figure — Il est important de comprendre où vient se placer la figure produite par l'environnement `tikzpicture`. La boîte englobante le dessin produit est simplement mise dans la continuité du texte en cours comme le montre l'exemple suivant.

On commence une phrase pour illustrer

```
\begin{tikzpicture}
  \draw (0,0) circle (1);
\end{tikzpicture}
```

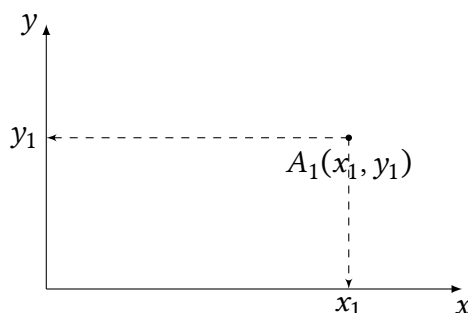
le placement de la `\lstinline{tikzpicture}` qui se place dans la continuité de la ligne du paragraphe en cours.

On commence une phrase pour illustrer  le placement de la `tikzpicture` qui se place dans la continuité de la ligne du paragraphe en cours.

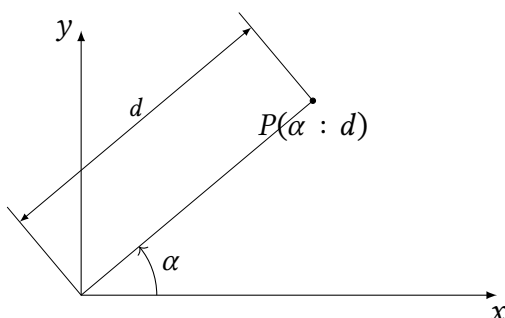
Il faudra donc souvent placer les figures produites par Tikz soit par un alignement de texte (en particulier l'environnement `center`), soit par l'inclusion dans l'environnement flottant `figure` (voir section 4.3).

7.2.3 Le repérage des points

COORDONNÉES CARTÉSIENNES : (x, y) . Les points du plan peuvent être repérés avec les coordonnées cartésiennes dans un repère orthonormé de vecteur de base de longueur 1 cm. L'axe des abscisses est dirigé vers la droite, et l'axe des ordonnées est dirigé vers le haut. La position d'un point est alors repérée par un couple de nombre (x, y) .



COORDONNÉES POLAIRES : (a : r). Les points du plan peuvent être repérés avec les coordonnées polaires. Pour représenter le point P , la syntaxe est (a, d) où a est l'angle orienté (en degrés) entre le vecteur de base des abscisses et le vecteur défini par \vec{OP} et d la distance OP .



Et l'origine? — L'origine n'a pas de position prédéfinie. En effet, la figure produite est une *boîte* encapsulant tous les éléments tracés. Cette boîte est placée par \TeX dans le PDF, et donc la position de l'origine dépendra du placement de la boîte et des éléments tracés sur le dessin. Mais aucune inquiétude car tous les éléments seront positionnés relativement à l'origine.

Tout cela reste bien abstrait, mais la section suivante va nous permettre d'illustrer le repérage de point en traçant des lignes.

Nommage des points — On pourra se définir des points avec des noms pour les manipuler plus simplement. Pour ce faire, on utilisera la commande :

`\coordinate (<nom>) at <point>`

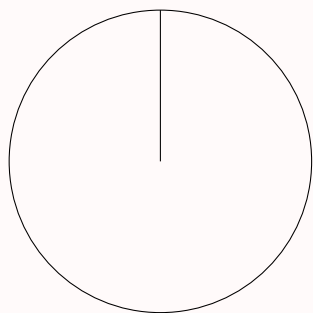
Comme vu précédemment, le $\langle point \rangle$ pour s'exprimer en coordonnées cartésiennes ou en polaires.

On pourra alors dessiner en appelant les points définis comme l'illustre l'exemple suivant.

```
\begin{tikzpicture}
\coordinate (B) at (90:2);
```



```
\coordinate (O) at (0,0);  
\draw (O)--(B);  
\draw (O) circle (2);  
\end{tikzpicture}
```



7.2.4 Dessiner et colorier

Pour dessiner et colorier, il y a deux commandes fondamentales de Tikz : `\draw` et `\fill`. Tant que nous n'avons pas vu les objets graphiques de Tikz, on ne va pas pouvoir montrer des exemples d'utilisation de ces commandes, mais ceci sera fait dans les exemples suivants.

Ces deux commandes, comme les nombreuses commandes Tikz, prennent un argument optionnel où l'on peut renseigner tout un tas de choses, souvent sous la forme `cle= valeur`. Dans cette introduction expresse à Tikz, nous n'allons pas décrire toutes les options ni comment on utilise ces deux commandes, il faudra comprendre leur utilisation à partir des exemples.

Couleurs

Grosueur de traits

7.2.5 Les lignes

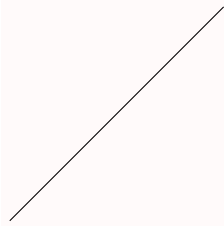
Une des principaux outils de dessin est la *ligne*. Simple segment, ligne brisée ou courbe, les lignes sont les briques de base de tout dessin. Nous allons donc voir comment tracer des lignes avec Tikz.

7.2.6 Le segment

La ligne la plus simple est sans doute celle qui relie deux points en ligne droite, autrement dit, le segment.

En Tikz, on relie deux points par une ligne droite avec la syntaxe `--`. De plus, on dessine un élément graphique avec la commande `\draw`, cette commande est d'ailleurs très utile avec Tikz.

```
\begin{tikzpicture}
  \draw (0,0)--(45:4);
\end{tikzpicture}
```

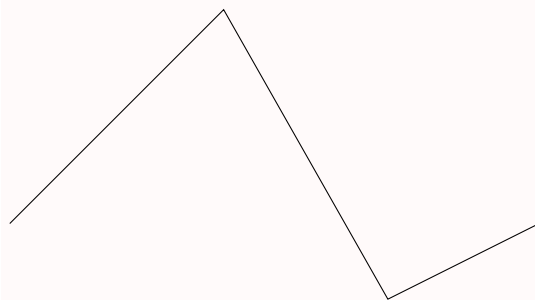


Dans cet exemple on relie donc le point défini en coordonnées cartésiennes ($x = 0, y = 0$) au point défini en coordonnées polaire $r = 4$ cm et $\theta = 45^\circ$.

Lignes brisées

Pour une ligne brisée, il suffit d'enchaîner les points en les reliant avec des lignes droites.

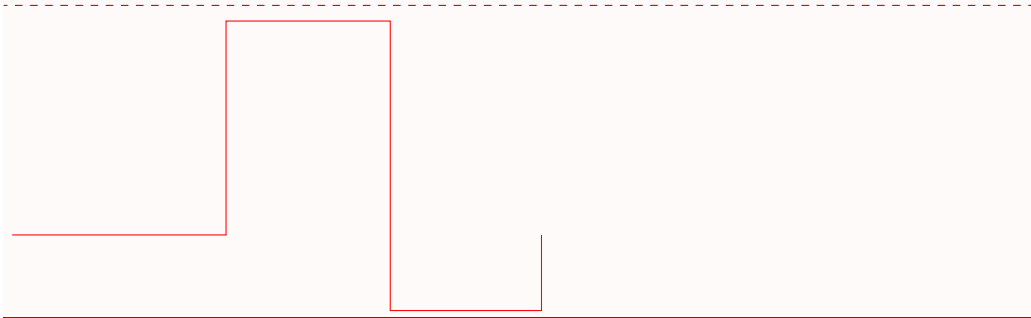
```
\begin{tikzpicture}
  \draw (0,0)--(45:4)--(5,-1)--(7,0);
\end{tikzpicture}
```



Il existe d'autre moyen de relier les points avec des lignes droites. On peut demander à Tikz de n'utiliser que des lignes horizontales et verticales avec la syntaxe `-|` ou `|-`.

```
\begin{tikzpicture}
```

```
\draw[color=red] (0,0)--(45:4)--(5,-1)--(7,0);
\end{tikzpicture}
```



Lignes courbes

Pour les lignes courbes, Tikz fournit une syntaxe spéciale. Le principe repose sur la notion mathématique des *courbes de Bézier cubique*. Pour ceci, on va relier deux points P_0 et P_3 avec une ligne courbe en spécifiant deux autres points¹ P_1 et P_2 , dits de contrôles. La courbe part alors de P_0 vers P_1 et arrivant en P_3 selon la direction $P_2 - P_3$. Les distances respectivement de P_1 à P_0 et de P_2 à P_3 vont permettre de régler si la courbe suit plus ou moins les directions $P_0 - P_1$ et $P_2 - P_3$. La figure 7.1 permet d'illustrer ce concept.

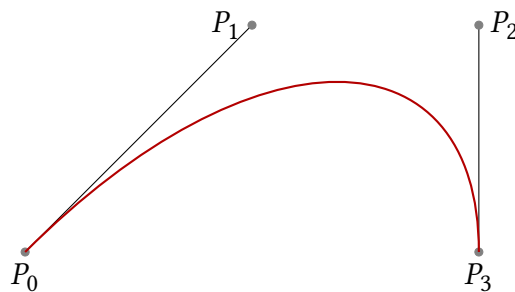


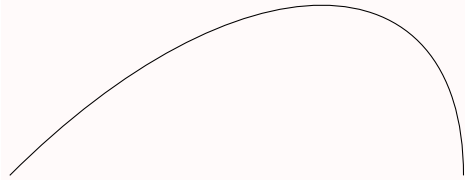
FIGURE 7.1 – Figure illustrant les courbes de Bézier cubiques.

Pour indiquer les points de contrôles, on utilisera la syntaxe suivante $(x_0, y_0) .. controls (x_1, y_1) and (x_2, y_2) .. (x_3, y_3)$ où les (x_i, y_i) sont les coordonnées des points P_i . Pour produire la courbe de Bézier précédente, il suffit alors de produire le code suivant.

```
\begin{tikzpicture}[scale=3]
\draw (0,0) .. controls (1,1) and (2,1) .. (2,0);
```

1. En réalité, on peut n'en spécifier qu'un seul.

```
\end{tikzpicture}
```

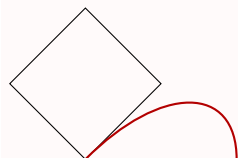


Coordonnées relatives

On pourra aussi construire des lignes avec des coordonnées relatives. La syntaxe utilise l'opérateur `++`. La notation `\draw (a,b)--++(x,y)`; sera alors équivalente à `\draw (a,b)--(c,d)`; avec $c = a + x$ et $d = b + y$.

On pourra aussi définir les points de contrôle pour les lignes courbes relativement aux points de départ et d'arrivée.

```
\begin{tikzpicture}
  \draw (0,0) -- ++(1,1) -- ++(-1,1) --++ (-1,-1) -- cycle;
  \draw[thick,darkred] (0,0) .. controls ++(1,1) and ++(0,1) ..
    (2,0);
\end{tikzpicture}
```



7.2.7 Formes géométriques

7.2.8 Ajouter de annotations

7.3 PGFPLOTS

7.4

PACKAGES UTILES

Dans ce chapitre, nous allons présenter quelques packages qui se révèlent utiles pour la production mathématique en général, de la rédaction d'un rapport, d'un photocopie, d'une publication, etc.

Nous ne présenterons pas les packages de manière exhaustive, mais plutôt par quelques exemples, que nous espérons, illustrerons correctement leur utilité, convaincront de leur utilité et vous ferons aller consulter leurs documentations.

\LaTeX est un logiciel libre avec une grande communauté d'utilisateurs et d'utilisatrices. Grâce à cela, et du fait de son mécanisme de *packages*, de nombreuses extensions du fonctionnement de base de \LaTeX ont été développées. Pour l'immense majorité des besoins classiques, pensez bien que quelqu'un-e a déjà eu ce besoin, et a sans doute développé ou stimulé le développement d'un package qui fournit de quoi satisfaire ces besoins.

8.1 LA GESTION DES COULEURS AVEC `xcolor`

Le package `xcolor`[21] permet de définir et d'utiliser des couleurs simplement. Pour cela, on pourra utiliser la commande suivante¹ :

```
\definecolor{<nom>}{<modele>}{<liste des coordonnees>}
```

Grâce à cette commande, on donnera un *<nom>* à la couleur, on spécifiera le *<modèle>* (rgb, RGB, cmyk, hsb, HTML, etc.) ainsi que la *<liste des coordonnées>* (qui

1. De très nombreuses façons de définir des couleurs existent, des ensembles de couleurs, des alias, etc. N'hésitez pas à parcourir la documentation pour explorer tous ces outils.

dépendra du modèle choisi). On pourra ainsi définir des couleurs :

```
\definecolor{rouge}{rgb}{0.7,0.1,0.1} % coordonnées entre 0 et 1
\definecolor{definition}{RGB}{56,97,146} % coordonnées entre 0
    et 255
\definecolor{nouvelle}{HTML}{#38924E} % couleur code HTML
```

8.1.1 Utiliser les couleurs

Une fois des couleurs accessibles, on peut colorer son document avec divers commandes. Sans exhaustivité, en voici quelques-unes :

```
\textcolor{couleur}{texte}
```

Cette commande permet de mettre du **texte en couleur**.

On peut aussi composer du texte dans une boîte colorée avec les deux commandes suivantes.

```
\colorbox{couleur}{texte}
```

```
\fcolorbox{couleur}{couleur}{texte}
```

L'exemple suivant illustre l'utilisation de ces trois commandes.

Voici `\textcolor{red}{du texte coloré}`. On peut aussi mettre les `\colorbox{red}{mots}` dans des `\fcolorbox{blue}{red}{boîtes}`.

Voici **du texte coloré**. On peut aussi mettre les **mots** dans des **boîtes**.

Il est possible aussi de changer la couleur de composition de façon pérenne (jusqu'au prochain changement) avec la commande suivante :

```
\color{couleur}
```

Il est possible aussi de changer la couleur de la page (là encore jusqu'au prochain changement de couleur de page) avec la commande suivante :

```
\pagecolor{couleur}
```

8.1.2 Charger des ensembles prédéfinis

Au chargement du package `xcolor`, on peut choisir quelques ensembles de couleurs prédéfinies. On aura toujours quelques couleurs de base qui sont toujours accessibles :

```
black, blue, brown, cyan, darkgray, gray, green, lightgray,
    lime, magenta, olive, orange, pink, purple, red, teal,
```

```
violet, white, yellow.
```

Autrement, la documentation du package `xcolor` [21], section 4, donne les différents ensembles de couleurs prédéfinies telles que `dvipsnames`, `svgnames` et `x11names`. Ces ensembles pourront être choisis au chargement du package. Par exemple, si l'on souhaite pouvoir utiliser les couleurs de l'ensemble `svgnames`, on chargera `xcolor` comme ceci :

```
\usepackage[svgnames]{xcolor}
```

préambule

8.2 RÉFÉRENCES CROISÉES AVEC `refcheck`

Le mécanisme de références croisées est très puissant et très pratique. Cependant, la rigueur de nommage des *label* est très difficile à tenir, surtout en tant que novices, et cela devient vite difficile de s'y retrouver.

De plus, on a tendance à beaucoup *labeliser* les éléments, notamment les équations, et donc à les composer en les numérotant (avec l'environnement `equation`). Or, on ne fait pas toujours référence à toutes les équations auxquelles on a ajouté un label, mais cela ne se repère pas facilement. Pour aider à nous y retrouver dans les labels et les références, le package `refcheck` [35] peut se révéler très utile. Ce package va indiquer s'il existe dans le texte une référence aux différents labels. Là encore, plutôt qu'un long discours, nous présentons ici un exemple illustrant l'utilité de ce package. Il faudra donc charger le package dans son préambule.

```
\usepackage{refcheck}
```

préambule

Une fois cela fait, le code suivant une fois compilé, produira le résultat présenté en figure 8.1. Les labels auxquels on ne fait pas appel dans le texte sont affublés des signes ? de chaque côté.

```
\begin{equation}\label{eq:test1}
  f(x)=x^{2}+2
\end{equation}
On teste~\eqref{eq:test1}

\section{Test}\label{sec:section}
```

```

\begin{equation}\label{eq:test3}
  f(x)=x^{2}+2
\end{equation}

\begin{equation}\label{eq:test2}
  f(x)=x^{2}+2
\end{equation}

see \ref{sec:section}

```

	$f(x) = x^2 + 2$	(1) eq:test1
On teste (1)		
1 Test		
(sec:section)	$f(x) = x^2 + 2$	(2) ?eq:test3?
	$f(x) = x^2 + 2$	(3) ?eq:test2?
see 1		

FIGURE 8.1 – Illustration de l’utilisation du package `refcheck`.

8.3 LES UNITÉS DU SYSTÈME INTERNATIONAL AVEC `siunitx`

Dès qu’on compose des mathématiques, et que celles-ci ont vocation à être un peu appliquées, il est souvent nécessaire de composer des nombres, des grandeurs et des unités. Les normes pour ces compositions sont complexes il y a beaucoup d’éléments à gérer : la gestion des espaces, des lettres, etc.

Pour composer correctement tout cela, nous disposons d’un package très utile : `siunitx`[49]. Celui-ci se charge dans le préambule par :

```
\usepackage{siunitx}
```

préambule

Ce package est extrêmement riche, et comme l’exemple suivant le montre, il permet de composer facilement des nombres tels que les nombres complexes (avec un i romain), des nombres en utilisant la notation scientifique (à partir de la notation informatique de ces nombres), et les unités du système international, etc.

Évidemment, il y a beaucoup plus de possibilités, et il faudra aller voir dans le document du package.

Soit $z = \text{complexnum}\{3+i5\}$ un nombre complexe.

Le calcul par la machine donne $\epsilon = \text{num}\{1.54e-13\}$

On peut utiliser les unités du système international : $\text{SI}\{6\}\{\text{m. kg}/(\text{s}^3.\text{A})\}$

Soit $z = 3 + 5i$ un nombre complexe.

Le calcul par la machine donne $\epsilon = 1,54 \times 10^{-13}$

On peut utiliser les unités du système international : 6 m kg/(s³ A)

Vous pourrez constater que la commande `\num` a transformé le `.` en `,` qui est le séparateur décimal en français (en anglais, on utilise le point). Cela s'est fait grâce à la commande du package `siunitx` suivante :

```
\sisetup{locale = FR}
```

préambule

8.4 PERSONNALISER LES LISTES AVEC `enumitem`

Le package `enumitem` [26] permet de modifier l'apparence des listes (`enumerate`, `itemize` et `description`). Là, encore, le package est très riche en possibilités, et il faudra aller étudier sa documentation pour en saisir toutes les subtilités. Ici, nous ne donnerons qu'un simple aperçu de son utilisation.

Dans le préambule, il faudra donc charger le package classiquement.

```
\usepackage{enumitem}
```

préambule

Nous présenterons dans la suite quelques exemples où les modifications sont *locales*, c'est-à-dire que la modification du comportement se change à l'utilisation de l'environnement, mais cela peut être fait de façon globale dans la préambule du document. Nous renvoyons à la documentation du package.

8.4.1 Personnalisation de `itemize`

La modification de l'environnement `itemize` consiste principalement au changement des tirets (fonctionnement par défaut lorsque le document est configuré pour le français).

```
Un peu de tendresse avec un changement de \lstinline+itemize+ :
\begin{itemize}[label=$\heartsuit$]
\item la réponse D ;
\item ne pas en abuser ;
\item 42.
\end{itemize}
```

```
Un peu de tendresse avec un changement de itemize :
♥ la réponse D ;
♥ ne pas en abuser ;
♥ 42.
```

8.4.2 Personnalisation de `enumerate`

Pour changer le type de numérotation, il suffit là encore de modifier le `label`. Il y a plusieurs styles disponibles :

- `\Roman*` qui produit I, II, ...;
- `\roman*` qui produit i, ii, ...;
- `\Alph*` qui produit A, B, C, ...;
- `\alph*` qui produit a, b, c, ...;
- `\arabic*` qui produit 1, 2, 3, ...

```
\begin{enumerate}[label=\Roman*]
\item c'est la première lettre ;
\item la deuxième ;
\item etc.
\end{enumerate}
```

```
I c'est la première lettre ;
II la deuxième ;
III etc.
```

On observe dans l'exemple précédent, qu'il n'y a pas de signe de ponctuation qui sépare le numéro de l'énumération du texte. Le mécanisme permet d'*enrober* le numéro de tout ce que l'on souhaite comme le montre l'exemple suivant.

```

\begin{enumerate}[label=(\arabic*) -- ]
\item mon premier est \dots
\item mon second est \dots
\item mon tout est \dots
\end{enumerate}

```

- (1) – mon premier est ...
(2) – mon second est ...
(3) – mon tout est ...
-

Plus on rajoute de caractères plus l'espace alloué aux numéros ne va plus être suffisant. Pour remédier à ce problème, il est tout à fait possible de les modifier. Là encore, cela dépasse le cadre de ce document et nous renvoyons à la documentation, ou au site [38] pour plus de détails.

8.4.3 Personnalisation de description

Pour la modification du style des descriptions, on utilisera la clé font (qui est aussi utilisable dans les environnements `enumerate` et `itemize`). Il suffit par cette clé de spécifier le style que l'on souhaite pour la description.

```

\begin{description}[font=\scshape\bfseries]
\item[Test] Ceci est un texte.
\end{description}

```

TEST Ceci est un texte.

8.5 DES BEAUX ENCADREMENTS AVEC `tcolorbox`

La composition de document avec \LaTeX a tendance à être de base très sobre. Il peut être intéressant de mettre en valeur des éléments du document avec des cadres, et de la couleur. Il existe un grand nombre de package qui permettent cela. Ici nous présentons à travers un unique exemple, l'utilisation du package `tcolorbox`.

```

\usepackage{tcolorbox}

```


préambule

Il existe de centaines d'options, de modifications possibles, de personnalisations, etc. La documentation du package [43] est immense, mais est très bien faite.

Nous nous limiterons à deux exemples : le fonctionnement par défaut, et un petit exemple de personnalisation.

```
\begin{tcolorbox}
  Contenu de la boîte.
\end{tcolorbox}

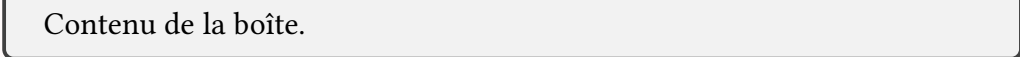
\begin{tcolorbox}[title=Titre de la boîte]
  Contenu de la boîte.
\end{tcolorbox}
```



Contenu de la boîte.



Titre de la boîte



Contenu de la boîte.

Grâce aux options du package on peut modifier quasiment tout de ces boîtes comme le montre l'exemple suivant.

```
\begin{tcolorbox}[
  rightrule=3mm,
  colback=red!5!white,
  colframe=red!75!black,
  arc=0mm,
  after title={\hfill\colorbox{white}{\color{black}approved}},
  title={\bfseries Titre de la boîte}
]
This is a \textbf{tcolorbox}.
\end{tcolorbox}
```



Titre de la boîte

approved

This is a **tcolorbox**.

8.6 FACILITER LA COMPOSITION MATHÉMATIQUE AVEC `tdsfrmath`

Citons la documentation [14] : *Le but de cette extension est de fournir des macros prêtes à l'usage à des professeurs de mathématiques des collèges, lycées – et plus si affinités – qui voudraient bien utiliser \LaTeX sans trop mettre le nez dans la programmation...*

Ce package peut aussi être utilisée par des étudiants et étudiantes dans le supérieur. Il fournit bon nombre de macros qui permettent de se faciliter la vie. Ce package se charge comme ceci :

```
\usepackage{tdsfrmath}
```

préambule

Voici un petit extrait des commandes fournies par le package.

On peut facilement composer les n -uplets :

```
\nuplet{a c s d}
```

De même, pour la composition des sous-ensembles de \mathbb{R} (ou des autres ensembles), de nombreuses options sont accessibles :

```
[\mathbb{R}^{*+} \quad \mathbb{R}^{-m}[Y].\]
```

Les intervalles s'écrivent très facilement : $\int x y$.
De même que les intégrales, les dérivées et les dérivées partielles:

```
[\intgen{2}{5}{\frac{\mathrm{D} x}{x^2}},  
 \quad \mathrm{derpart}\{f(x,y,z)\}{xyyz},  
 \quad \mathrm{drv}\{f(x)\}{x}.\]
```

On peut facilement composer les n -uplets : (a, c, s, d) .

De même, pour la composition des sous-ensembles de \mathbb{R} (ou des autres ensembles), de nombreuses options sont accessibles :

$$\mathbb{R}_+^* \quad \mathbb{R}_m[Y].$$

Les intervalles s'écrivent très facilement : $]x, y]$. De même que les intégrales, les dérivées et les dérivées partielles :

$$\int_2^5 \frac{dx}{x^2}, \quad \frac{\partial^6 f(x, y, z)}{\partial x^2 \partial y^3 \partial z}, \quad \frac{df(x)}{dx}.$$

8.7 LES DÉRIVÉES AVEC `derivative`

Le package `derivative` fournit de nombreuses commandes permettant de composer les dérivées de toutes sortes ainsi que les différentielles. Il permet de plus de définir simplement de nouveaux opérateurs de dérivée et de différentiation. On pourra aussi redéfinir simplement le style des opérateurs pour adapter l’affichage aux différentes conventions.

On changera classiquement ce package :

```
\usepackage{derivative}
```

préambule

On pourra alors par exemple utiliser les commandes illustrées dans l’exemple suivant :

```
\[\pdv[order={n+1,2}]{f}{x,y},\quad
\odif[style-var=single,
order={n,1,2}]{x,y,z,t}
,\quad \pdv{f}{x,y}_{(x_0,y_0)}
,\quad \odv{f}{x}
\]
```

$$\frac{\partial^{n+3} f}{\partial x^{n+1} \partial y^2}, \quad d_{x,y,z,t}^{n,1,2}, \quad \left(\frac{\partial^2 f}{\partial x \partial y} \right)_{(x_0,y_0)}, \quad \frac{df}{dx}$$

8.8 PERSONNALISER LES THÉORÈMES `thmtools`

Le package `thmtools` [41] permet de définir et personnaliser des environnements pour la composition de théorèmes, définitions, propositions, remarques, etc.

Là encore, les possibilités sont énormes et nous n’allons que présenter quelques exemples, principalement tirés de la documentation du package, qui les illustrent un tant soi peu.

Il faut charger le package :

```
\usepackage{amsthm} % pas besoin si mathtools
\usepackage{thmtools}
```

préambule

Avec ceci, on peut définir ses environnements dans *le préambule du document*.

```
\declaretheorem[numberwithin=section,name=Théorème]{theoremS}
```

```
\begin{theoremS}[Euclid]
  For every prime  $p$ , there is a prime  $p' > p$ .
  In particular, there are infinitely many primes.
\end{theoremS}
```

Théorème 8.8.1 (Euclid). *For every prime p , there is a prime $p' > p$. In particular, there are infinitely many primes.*

Quasiment tout est, là encore, personnalisable. Par exemple, on peut définir des styles dans le préambule :

```
\declaretheoremstyle[
  spaceabove=6pt, spacebelow=6pt,
  headfont=\normalfont\bfseries\color{red},
  notefont=\mdseries, notebraces={({})},
  bodyfont=\itshape,
  postheadspace=1em,
  qed=\qedsymbol
]{mystyle}
\declaretheorem[style=mystyle, name=Théorème]{styledtheorem}
```

qui permet de produire le résultat suivant.

```
\begin{styledtheorem}[Euclid]
  For every prime  $p$ \dots
\end{styledtheorem}
```

Théorème 1 (Euclid). *For every prime p ...* □

Ce package permet notamment de faciliter les références aux environnements. Nous renvoyons à la documentation pour plus de détail.

On peut produire très facilement la liste des théorèmes que l'on a utilisés grâce à la commande suivante.

```
\listoftheorems
```

8.9 COMPOSITION AVANCÉE DES TABLEAUX

Nous n'avons exploré qu'une petite partie de la composition des tableaux avec \LaTeX . Dès que les tableaux deviennent un peu trop compliqués, alors le mécanisme de base montré en section 3.9.6 ne suffit plus. Plusieurs packages sont utiles, mais nous ne rentrerons pas dans les détails.

array : ce package fournit des extensions à l'utilisation de l'environnement `tabular`, en particulier, des gestions de formatage de colonne avec la possibilité de définir de nouveau type de colonne [34].

tabularx : ce package, basé sur **array**, fournit en plus du formatage de colonne facilité, l'environnement `tabularx` qui permet de spécifier la largeur totale du tableau [10].

longtable : ce package permet de composer des tableaux sur plusieurs pages [8].

booktabs : ce package fournit des commandes supplémentaires qui permette d'améliorer l'esthétique de la composition des tableaux, notamment des filets horizontaux [11].

tabularray : ce nouveau (2021) package [28] est remarquable et est sans doute une révolution² dans les outils pour les tableaux en \LaTeX . Ce package sépare complètement le contenu du style du tableau, ancrant davantage \LaTeX dans la séparation fond/forme. De plus, le style du tableau peut être entièrement défini par le système de clé/valeur. L'exemple suivant illustre la puissance de ce package.

```
% couleurs utilisées : fournies par le package 'ninecolors'

\begin{tblr}{width=0.8\textwidth,
  colspec={l X[2] X[3] X[-1]}, % spécification des
  colonnes
  row{odd} = {bg=azure8}, % lignes impaires
  % première ligne
  row{1} = {bg=azure3, fg=white, font=\sffamily,c},
  % première colonne
  column{1} = {bg=purple7},
  % filets horizontaux
  hlines = {1pt,white},
```

2. En effet, au lieu de modifier les commandes de table internes de \LaTeX , comme les font classiquement les autres packages du domaine, l'auteur a abandonné l'utilisation des commandes \TeX de base et a utilisé les fonctions $\LaTeX3$, langage désormais mature, pour analyser le tableau puis le composer.


```

}
Alpha & Beta & Gamma & Delta \\
Epsilon & Zeta & Eta & Theta \\
Iota & Kappa & Lambda & Mu \\
Nu & Xi & Omicron & Pi \\
Rho & Sigma & Tau & Upsilon \\
Phi & Chi & Psi & Omega \\
\end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi
Rho	Sigma	Tau	Upsilon
Phi	Chi	Psi	Omega

8.10 UTILISER LE PRINCIPE DES TODO LIST AVEC **todonotes**

Lors de travaux à plusieurs, ou bien de long travaux qui nécessitent de noter des choses pour ne pas les oublier, il est souvent nécessaire de se faire des *TODO list*, comme il est d'usage chez les développeurs ou développeuses. Il s'agira en fait ici d'annoter le document avec des remarques pour ce qu'il reste à faire.

Le package **todonotes** [31] est fait pour cela :

```
\usepackage[colorinlistoftodos,french]{todonotes}
```

préambule

Le code suivant :

```
\section{\todo[author=Donald]{Titre à changer}Section}
```

```
\todo[inline,author=Maxime,color=blue!30]{Section à écrire}
```

```
\missingfigure{Figure d'illustration}
```

Un paragraphe qui ne veut pas dire grand chose et qu'il va falloir améliorer, ça c'est sûr. On peut même faire des erreurs de `\todo[author=Troisième,color=green!30]{Changer ces guillemets}'` et mettre le changement en TODO.

`\listoftodos`

produit alors le résultat montré en figure 8.2.

1 **Section**

Maxime: Section à écrire

Figure manquante Figure d'illustration

An paragraphe qui ne veut pas dire grand chose et qu'il va falloir améliorer, ça c'est sur. On peut même faire des erreurs de "typo" et mettre le changement en TODO.

Liste des points à traiter

Donald	Titre à changer	1
Section à écrire	Figure : Figure d'illustration	1
Troisième	Changer ces guillemets	1

FIGURE 8.2 – Résultat de l'utilisation du package todonotes.

8.11 FAIRE DES FEUILLES D'EXERCICES AVEC **exesheet**

Dès que l'on enseigne les mathématiques, on a besoin de faire des feuilles d'exercices, des examens, des corrections, etc. Antoine Missier, un francophone, a produit le package **exesheet** [33] très utile qui définit de nombreuses commandes

qui facilitent grandement la vie lorsqu'on produit un document contenant des exercices (sectionnement d'exercice, environnements de questions, d'énumérations, etc.).

Ce package permet de composer des exercices ainsi que leur corrections qui suivant l'option `display` fournie au chargement, s'affiche ou non.

De plus, ce package permet d'annoncer la correction, et de détailler le barème.

On peut aussi produire des exercices en laissant de la place pour que les étudiants et les étudiantes rédigent leur réponse dans un espace contraint en dessous de la question.

On pourra charger le package comme ceci³ :

```
\usepackage{exesheet} % 'naffiche pas la correction
\usepackage[output=questions|answers|both]{exesheet} % pour
    choisir ce que 'lon affiche, par défaut les exercices et les
    réponses
\usepackage[display=none|pts|notes]{exesheet} % si dans les
    corections on souhaite faire figurer les points ou les points
    et les notes
\usepackage[display=notes,marginwidth=unset]{exesheet} % le
    package modifie les marges du documents 'lorsquon affiche des
    points ou les notes, 'loption marginwidth permet 'dajuster
    ou 'dempêcher cela
```

préambule

Pour illustrer par l'exemple l'utilisation de ce package, le code suivant, avec en préambule `\usepackage[display=notes,marginwidth=unset]{exesheet}`, permet de produire le résultat visible en figure 8.3.

```
\exercise[\totalexe{4}] \label{totalexe}
\begin{questions}
For each following question, say if the assertion is true or
false.
Justify the answer carefully.
\begin{tablenum1}
\item $ \int_0^{\sqrt{3}} \frac{1}{x+\sqrt{3}} \, dx = \ln 2$,
\item $ \int_2^e \frac{1}{x \ln x} \, dx = -\ln 2$,
```

3. On ne présente ici que les options qui nous semblent principales, mais il en existe de nombreuses autres

```

\item* The function  $F$  defined on  $\mathbf{R}$  by
 $F(x) = \int_0^x \frac{1}{t^2+t+1} dt$  is
    increasing on  $\mathbf{R}$ .
\end{tablenum1}
\end{questions}

\begin{answers}
\begin{enumerate}
\item \note[1]{0,5 for the anti-derivative \0,5 for
    simplification of  $\ln$ }
    We calculate:
    \[ \int_0^{\sqrt{3}} \frac{1}{x+\sqrt{3}} dx
    = \big[ \ln \left( x + \sqrt{3} \right) \big]_0^{\sqrt{3}}
    = \ln \left( 2\sqrt{3} \right) - \ln \sqrt{3}
    = \ln \left( \frac{2}{\sqrt{3}} \right)
    = \ln 2.
    \]
    \textbf{TRUE}.

\item \note[1.5]{1 for the anti-derivative \0.5 for the
    integral value \[2ex]
    we could also see that  $\frac{1}{x} \ln x > 0$  on  $[2, \mathbf{e}]$ 
    while  $-\ln 2 < 0$ 
    }
    We have  $\frac{1}{x} \ln x = \frac{(\frac{1}{x})' \ln x + \frac{1}{x} (\ln x)'}{(\frac{1}{x})^2}$ 
    with  $u(x) = \ln x$ , which is positive on  $[2, \mathbf{e}]$ .
    Hence

    \[ \int_2^{\mathbf{e}} \frac{1}{x} \ln x dx
    = \Big[ \ln(\ln x) \Big]_2^{\mathbf{e}}
    = \ln(\ln \mathbf{e}) - \ln(\ln 2) = \ln 1 - \ln(\ln 2) = -\ln(\ln 2).
    \]

    \textbf{FALSE}.

\item \note*{1.5}
    The function  $F$  defined on  $\mathbf{R}$  by
    \[ F(x) = \int_0^x \frac{1}{t^2+t+1} dt \]

```

4 pts

Exercise 1

For each following question, say if the assertion is true or false. Justify the answer carefully.

1. $\int_0^{\sqrt{3}} \frac{1}{x+\sqrt{3}} dx = \ln 2$,
2. $\int_2^e \frac{1}{x \ln x} dx = -\ln 2$,
3. The function F defined on \mathbf{R} by $F(x) = \int_0^x \frac{1}{t^2+t+1} dt$ is increasing on \mathbf{R} .

Correction

1
0,5 for the
anti-derivative
0,5 for simpli-
fication of ln

1. We calculate:

$$\int_0^{\sqrt{3}} \frac{1}{x+\sqrt{3}} dx = \left[\ln(x+\sqrt{3}) \right]_0^{\sqrt{3}} = \ln(2\sqrt{3}) - \ln\sqrt{3} = \ln\left(\frac{2\sqrt{3}}{\sqrt{3}}\right) = \ln 2.$$

TRUE.

1.5
1 for the anti-
derivative
0.5 for the in-
tegral value

2. We have $\frac{1}{x \ln x} = \frac{\frac{1}{x}}{\ln x} = \frac{u'(x)}{u(x)}$ with $u(x) = \ln x$, which is positive on $[2, e]$. Hence

$$\int_2^e \frac{1}{x \ln x} dx = \left[\ln(\ln x) \right]_2^e = \ln(\ln e) - \ln(\ln 2) = \ln 1 - \ln(\ln 2) = -\ln(\ln 2).$$

FALSE.

we could also see
that $\frac{1}{x \ln x} > 0$
on $[2, e]$ while
 $-\ln 2 < 0$

3. The function F defined on \mathbf{R} by

$$F(x) = \int_0^x \frac{1}{t^2+t+1} dt$$

1.5

0.5 for F'
1 for the sign of
 F' and conclusion

is derivable on \mathbf{R} and its derivative is such that $F'(x) = \frac{1}{x^2+x+1}$. The denominator is a quadratic polynomial, always positive because its discriminant is $\Delta = -3 < 0$. Thus F is increasing on \mathbf{R} .

TRUE.

FIGURE 8.3 – Exemple de composition d'un exercice, de sa corrections et des annotations en utilisant le package `exosheet`.

```

is derivable on  $\mathbf{R}$  and its derivative is such that
 $F'(x) = \frac{1}{x^2+x+1}$ .
\note{0.5 for  $F'$  1 for the sign of  $F'$  and conclusion}
The denominator is a quadratic polynomial, always positive
because its discriminant is
 $\Delta = -3 < 0$ .
Thus  $F$  is increasing on  $\mathbf{R}$ .
\textbf{TRUE}.
\end{enumerate}
\end{answers}

```

8.12 ÉCRIRE DU CODE INFORMATIQUE AVEC listings

Pour utiliser `listings`[15], il faudra le charger dans son préambule :

```
\usepackage{listings}
```

préambule

8.12.1 Utilisation standard

Une fois cela chargé, on pourra utiliser l'environnement `lstlisting`.

```
\begin{lstlisting}
Le code à mettre ici.
\end{lstlisting}
```

On spécifiera le langage du code à afficher via la clé `language=<langage>`. Nous renvoyons à la documentation de `listings` [15] pour la liste complète des très nombreux langages supportés.

Pour composer un code Python, il suffit alors d'écrire :

```
\begin{lstlisting}[language=python]
def somme(a,b):
    return a+b

c = somme(4,5)
\end{lstlisting}
```

La figure 8.4 montre le résultat de ce code.

```
def somme(a,b):
    return a+b

c = somme(4,5)
```

FIGURE 8.4 – Résultat d'un simple code Python avec le package `listings`

Il est aussi possible de composer le code d'un fichier externe avec la commande :

```
[macros]{\lstinputlisting@}\lstineline+\lstinputlisting+}
\lstinputlisting[language=python]{script.py}
```

La commande `\lstinputlisting` permet même de spécifier les lignes de début et de fin du fichier que l'on souhaite afficher en utilisant les options `firstline=<nbr>` et `endline=<nbr>`.

L'environnement `lstlisting` et la commande `\lstinputlisting` produisent des *flottants* dès que l'on ajoute comme option `caption=<titre du code>`. Pour y faire référence, on utilisera l'option `label=<clé>`.

```
\begin{lstlisting}[caption={Titre de mon code}, label=code:
  exemple]
...
\end{lstlisting}
```

On peut aussi écrire du code en *mode texte*, c'est-à-dire dans le texte courant avec la commande `\lstinline`. Pour délimiter le contenu à composer en ligne, on utilisera un caractère qui ne figure pas de le code, comme l'illustre l'exemple suivant.

```
Les commandes du package \texttt{listings} vues jusque là sont \
  \lstinline+\lstinputlisting+ et \lstinline!\lstinline!. Nous
  allons bientôt voir \lstinline$\lstset$.
```

Les commandes du package `listings` vues jusque là sont `\lstinputlisting` et `\lstinline`. Nous allons bientôt voir `\lstset`.

La commande `\lstinline` permet aussi de spécifier le langage en option.

8.12.2 Personnalisation

La figure 8.4 montre que le style par défaut de composition du code n'est pas forcément très agréable à lire. En effet, il est désormais courant de montrer le code écrit comme dans les éditeurs de texte, avec notamment des couleurs et une fonte à chasse fixe.

Pour paramétrer le langage utilisé par défaut, ainsi que le style, on utilisera la commande `\lstset`.

Nous n'allons pas détailler les nombreuses possibilités de personnalisation, et nous nous contenterons de donner un exemple donnant plusieurs pistes de personnalisation.

```
\definecolor{codegreen}{rgb}{0,0.6,0}
\definecolor{codegray}{rgb}{0.5,0.5,0.5}
\definecolor{codepurple}{rgb}{0.58,0,0.82}
\definecolor{backcolour}{rgb}{0.95,0.95,0.92}
\lstset{
```

```

backgroundcolor=\color{backcolour},
commentstyle=\color{codegreen},
keywordstyle=\color{magenta},
numberstyle=\tiny\color{codegray},
stringstyle=\color{codepurple},
basicstyle=\ttfamily\footnotesize,
breakatwhitespace=false,
breaklines=true,
numbers=left,
numbersep=5pt,
language=Python,
}
\begin{lstlisting}
def somme(a,b):
    return a+b

c = somme(4,5)
\end{lstlisting}

```

```

1 def somme(a,b):
2     return a+b
3
4 c = somme(4,5)

```

Nous conseillons de créer des *styles* avec la commande `\lstdefinestyle` et de sélectionner un style avec `\lstset` comme l'illustre l'exemple suivant.

```

\definecolor{codegreen}{rgb}{0,0.6,0}
\definecolor{codegray}{rgb}{0.5,0.5,0.5}
\definecolor{codepurple}{rgb}{0.58,0,0.82}
\definecolor{backcolour}{rgb}{0.95,0.95,0.92}
\lstdefinestyle{pythonStyle}{
backgroundcolor=\color{backcolour},
commentstyle=\color{codegreen},
keywordstyle=\color{magenta},
numberstyle=\tiny\color{codegray},
stringstyle=\color{codepurple},
basicstyle=\ttfamily\footnotesize,
breakatwhitespace=false,
breaklines=true,

```



```

numbers=left,
numbersep=5pt,
language=Python,
}
\lstset{style=pythonStyle}
\begin{lstlisting}
def somme(a,b):
    return a+b

c = somme(4,5)
\end{lstlisting}

```

```

1 def somme(a,b):
2     return a+b
3
4 c = somme(4,5)

```

Ce mécanisme vous permettra de changer facilement de style, de langage, etc.

8.13 METTRE EN PAGE DES PSEUDO CODES

De nombreux packages permettent de mettre en forme des pseudo codes.

8.13.1 Au début, `algorithms`

En 1994, le package `algorithms` [7] a fourni deux environnements : `algorithm` et `algorithmic`. L'environnement `algorithm` permet de créer des flottants (différents de figure et table, voir 4.3) dans lesquels on peut insérer du pseudo code généré en utilisant l'environnement `algorithmic`.

Pour utiliser uniquement l'environnement `algorithm`, comme nous le conseillons après, on chargera :

```
\usepackage{algorithm}
```

préambule

8.13.2 Le pseudo code

`algorithmicx` et `algpseudocodex`. — L'environnement `algorithmic` a sans doute un peu standardisé la mise en page et les commandes pour le pseudo code. Le package `algorithmicx` [18] (dont la francisation se fait en utilisant

`frpseudocode` [17]) est venu étendre les fonctionnalités de cet environnement. Il a été ensuite de nouveau étendu par `algpseudocodex` [29] que nous conseillons d'utiliser, même si ce dernier ne fournit pas de version française des mots clés.

Ainsi, on pourra utiliser `algorithm` et `algpseudocodex` de la façon suivante :

```
\begin{algorithm}
\begin{algorithmic}[1]
\Procedure{Euclide}{$a,b$}\Comment{Le p.g.c.d. de $a$ et $b$}
\State $r\gets a\bmod b$
\While{$r\mathbf{not}=0$}\Comment{On option le résultat si $r$ vaut 0}

\State $a\gets b$
\State $b\gets r$
\State $r\gets a\bmod b$
\EndWhile\label{euclidendwhile}
\State \Return $b$\Comment{Le p.g.c.d. est $b$}
\EndProcedure
\end{algorithmic}
\caption{'L'algorithme 'dEuclide}\label{euclid}
\end{algorithm}
```

code qui produira le résultat de l'algorithme 1.

Algorithm 1 L'algorithme d'Euclide

```
1: procédure EUCLIDE( $a, b$ ) ▷ Le p.g.c.d. de  $a$  et  $b$ 
2:    $r \leftarrow a \bmod b$ 
3:   tant que  $r \neq 0$  faire ▷ On option le résultat si  $r$  vaut 0
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   retourner  $b$  ▷ Le p.g.c.d. est  $b$ 
```

En français. — Vous pouvez constater que les mots clés sont ici en français car nous avons ajouté le code suivant dans notre préambule :

```
\usepackage{algorithm, algpseudocodex}
\algnewcommand\algorithmicend{\textbf{fin}}
\algnewcommand\algorithmicdo{\textbf{faire}}
```

préambule

```
\algnewcommand\algorithmicwhile{\textbf{tant que}}
\algnewcommand\algorithmicfor{\textbf{pour}}
\algnewcommand\algorithmicforall{\textbf{pour tous}}
\algnewcommand\algorithmicloop{\textbf{boucle}}
\algnewcommand\algorithmicrepeat{\textbf{répéter}}
\algnewcommand\algorithmicuntil{\textbf{' jusqu'à}}
\algnewcommand\algorithmicprocedure{\textbf{procédure}}
\algnewcommand\algorithmicfunction{\textbf{fonction}}
\algnewcommand\algorithmicif{\textbf{si}}
\algnewcommand\algorithmicthen{\textbf{alors}}
\algnewcommand\algorithmicelse{\textbf{sinon}}
\algnewcommand\algorithmicrequire{\textbf{Données :}}
\algnewcommand\algorithmicensure{\textbf{Demandé :}}
\algnewcommand\algorithmicreturn{\textbf{retourner}}
\algnewcommand\algorithmicoutput{\textbf{sortie}}
```

préambule

8.13.3 **algorithm2e**

Le **algorithm2e** [12] est package complètement différent. Il a été créé par un francophone, il fournit les mots clés dans de nombreuses langues et ne nécessite pas l'usage de **algorithm** puisqu'il fournit directement de quoi gérer les flottants.

BIBLIOGRAPHIE

9.1 LES LIVRES PAPIERS OU PDF

- [1] Jacques ANDRÉ. *Petites leçons de typographie*. <http://jacques-andre.fr/faqtypo/lessons.pdf>. 2017.
- [2] Marc BAUDOIN. *Apprends L^AT_EX!* 2012. URL : <https://www.ctan.org/tex-archive/info/apprends-latex/>.
- [6] Denis BITOUZÉ et Jean-Côme CHARPENTIER. *Introduction à L^AT_EX*. Pearson, juill. 2010. ISBN : 2-7440-7451-9. URL : <http://www.pearson.fr/livre/?GCOI=27440100259650>.
- [22] Donald E. KNUTH. *Le T_EXbook : composition informatique*. Trad. par Jean-Côme CHARPENTIER. Vuibert informatique. Vuibert, 2003. ISBN : 9782711748198.
- [24] Leslie LAMPORT. *L^AT_EX : a document preparation system*. Anglais. Second. Addison-Wesley, 1994. ISBN : 0-201-52983-1. URL : <http://www.informit.com/store/product.aspx?isbn=0201529831>.
- [40] Maïeul ROUQUETTE, Enimie ROUQUETTE et Brendan CHABANNES. *(Xe)L^AT_EX appliqué aux sciences humaines*. Tampere : Atramenta. 268 p. ISBN : 978-952-273-073-2.
- [47] Gérard TISSEAU et Jacques DUMA. *Tikz pour l'impatient*. <http://math.et.info.free.fr/TikZ/index.html>. 2017.

9.2 LES LIENS INTERNETS

- [4] Denis BITOUZÉ. *Conférence 1 : les bases de L^AT_EX*. 2020. URL : <https://gte.univ-littoral.fr/Members/denis-bitouze/pub/latex/diapositives-cours-d/conference-n-1/@@download/file/en-ligne1.pdf>.
- [5] Denis BITOUZÉ. *Site web*. 2020. URL : <http://gte.univ-littoral.fr/members/dbitouze/pub>.
- [23] *L'association GUTenberg*. 2024. URL : <https://gutenberg-asso.fr>.
- [36] *T_EXnique*. 2020. URL : <https://texnique.fr/osqa/>.
- [38] OPEN-FREAX. *Personnaliser ses listes L^AT_EX avec enumitem*. 2013. URL : <https://open-freax.fr/listes-latex-enumitem/> (visité le 25/09/2020).
- [39] Maïeul ROUQUETTE. *BibL^AT_EX and biber*. 2019. URL : <https://geekographie.maieul.net/211>.
- [42] *T_EX Stackexchange*. 2020. URL : <https://tex.stackexchange.com>.

9.3 LES DOCUMENTATIONS

- [3] Claudio BECCARI et Enrico GREGORIO. *The imakeidx package. A package for producing multiple indexes*. Version 1.3e. 15 oct. 2016. URL : <https://ctan.org/pkg/imakeidx>.
- [7] Rogério BRITO et Peter WILLIAMS. *The algorithms package. A suite of tools for typesetting algorithms in pseudo-code*. Version 0.1. 9 oct. 2018. URL : <https://ctan.org/pkg/algorithms>.
- [8] David CARLISLE, David KASTRUP et THE L^AT_EX PROJECT TEAM. *The longtable package. Allow tables to flow over page boundaries*. Version 4.19. 1^{er} nov. 2023. URL : <https://ctan.org/pkg/longtable>.
- [9] David CARLISLE et THE L^AT_EX PROJECT TEAM. *The graphicx package. Enhanced support for graphics*. Version 1.2d. 12 nov. 2021. URL : <https://ctan.org/pkg/graphicx>.
- [10] David CARLISLE et THE L^AT_EX PROJECT TEAM. *The tabularx package. Tabulars with adjustable-width columns*. Version 2.11c. 1^{er} nov. 2023. URL : <https://ctan.org/pkg/tabularx>.
- [11] Daniel ELS et Simon FEAR. *The booktabs package. Publication quality tables in L^AT_EX*. Version 1.61803398. 14 jan. 2020. URL : <https://ctan.org/pkg/booktabs>.

- [12] Christophe FIORIO et Ivan KOKAN. *The algorithm2e package. Floating algorithm environment with algorithmic keywords*. Version 5.2. 6 avr. 2020. URL : <https://ctan.org/pkg/algorithm2e>.
- [13] Daniel FLIPO. *The babel-french package. Babel contributed support for French*. Version 3.5q. 9 mars 2023. URL : <https://ctan.org/pkg/babel-french>.
- [14] Yvon HENEL. *The tdsfrmath package. Macros for French teachers of mathematics*. Version 1.3. 24 juin 2016. URL : <https://ctan.org/pkg/tdsfrmath>.
- [15] Jobst HOFFMANN, Brooks MOSES et Carsten HEINZ. *The listings package. Typeset source code listings using L^AT_EX*. Version 1.9. 6 oct. 2023. URL : <https://ctan.org/pkg/listings>.
- [16] Morten HØGHOLM et al. *The mathtools package. Mathematical tools to use with amsmath*. Version 1.29. 30 juin 2022. URL : <https://ctan.org/pkg/mathtools>.
- [17] Oliver IRWIN. *The frpseudocode package. French translation for the algorithmicx package*. Version 0.3.0. 10 août 2020. URL : <https://ctan.org/pkg/frpseudocode>.
- [18] Szász JÁNOS. *The algorithmicx package. The algorithmic style you always wanted*. 24 juin 2016. URL : <https://ctan.org/pkg/algorithmicx>.
- [19] Alan JEFFREY, Frank MITTELBACH et THE L^AT_EX PROJECT TEAM. *The inputenc package. Accept different input encodings*. Version 1.3d. 1^{er} juin 2021. URL : <https://ctan.org/pkg/inputenc>.
- [20] Roger KEHR et Joachim SCHROD. *The xindy package. A general-purpose index processor*. Version 2.5.1. 18 avr. 2017. URL : <https://ctan.org/pkg/xindy>.
- [21] Uwe KERN et THE L^AT_EX PROJECT TEAM. *The xcolor package. Driver-independent color extensions for L^AT_EX and pdfLaTeX*. Version 2.14. 12 juin 2022. URL : <https://ctan.org/pkg/xcolor>.
- [25] Philipp LEHMAN et Philip KIME. *The BibLaTeX package. Sophisticated Bibliographies in L^AT_EX*. Version 3.19. 6 mars 2023. URL : <https://ctan.org/pkg/biblatex>.
- [26] Javier Bezos LÓPEZ. *The enumitem package. Control layout of itemize, enumerate, description*. Version 3.9. 22 juin 2019. URL : <https://ctan.org/pkg/enumitem>.
- [27] Javier Bezos LÓPEZ et Johannes L. BRAAMS. *The babel package. Multilingual support for L^AT_EX, LuaL^AT_EX, XeL^AT_EX, and Plain T_EX*. Version 3.96. 25 oct. 2023. URL : <https://ctan.org/pkg/babel>.

- [28] Jianrui LYU. *The tabularray package. Typeset tabulars and arrays with \LaTeX 3*. Version 2023A. 1^{er} mars 2023. URL : <https://ctan.org/pkg/tabularray>.
- [29] Christian MATT. *The algpseudocodex package. Package for typesetting pseudocode*. Version 1.1.2. 19 avr. 2023. URL : <https://ctan.org/pkg/algpseudocodex>.
- [30] Wolfgang MAY et Andreas SCHEDLER. *The ntheorem package. Enhanced theorem environment*. Version 1.33. 18 avr. 2017. URL : <https://ctan.org/pkg/ntheorem>.
- [31] Henrik Skov MIDTIBY. *The todonotes package. Marking things to do in a \LaTeX document*. Version 1.1.6. 31 jan. 2023. URL : <https://ctan.org/pkg/todonotes>.
- [32] Vedran MILETIĆ et al. *The beamer package. A \LaTeX class for producing presentations and slides*. Version 3.70. 15 mai 2023. URL : <https://ctan.org/pkg/beamer>.
- [33] Antoine MISSIER. *The exesheet package. Typesetting exercise or exam sheets*. Version 2.4. 29 oct. 2023. URL : <https://ctan.org/pkg/exesheet>.
- [34] Frank MITTELBACH et THE \LaTeX PROJECT TEAM. *The array package. Extending the array and tabular environments*. Version 2.5g. 1^{er} nov. 2023. URL : <https://ctan.org/pkg/array>.
- [35] Oleg V. MOTYGIN. *The refcheck package. Check references (in figures, table, equations, etc)*. Version 1.9.1. 24 juin 2016. URL : <https://ctan.org/pkg/refcheck>.
- [37] Heiko OBERDIEK et al. *The hyperref package. Extensive support for hypertext in \LaTeX* . Version 7.01d. 31 oct. 2023. URL : <https://ctan.org/pkg/hyperref>.
- [41] Ulrich M. SCHWARZ et Yukai CHOU. *The thmtools package. Extensions to theorem environments*. Version 76. 4 mai 2023. URL : <https://ctan.org/pkg/thmtools>.
- [43] Thomas F. STURM. *The tcolorbox package. Coloured boxes, for \LaTeX examples and theorems, etc*. Version 6.1.0. 26 sept. 2023. URL : <https://ctan.org/pkg/tcolorbox>.
- [44] THE \LaTeX PROJECT TEAM. *The amsmath package. AMS mathematical facilities for \LaTeX* . Version 2.17o. 9 juin 2023. URL : <https://ctan.org/pkg/amsmath>.
- [45] THE \LaTeX PROJECT TEAM. *The fontenc package. Standard package for selecting font encodings*. 24 juin 2016. URL : <https://ctan.org/pkg/fontenc>.

- [46] THE L^AT_EX PROJECT TEAM. *The makeidx package. Standard L^AT_EX package for creating indexes.* Version 1.0m. 9 juin 2023. URL : <https://ctan.org/pkg/makeidx>.
- [48] Hideo UMEKI et David CARLISLE. *The geometry package. Flexible and complete interface to document dimensions.* Version 5.9. 31 oct. 2021. URL : <https://ctan.org/pkg/geometry>.
- [49] Joseph WRIGHT. *The siunitx package. A comprehensive (SI) units package.* Version 3.3.7. 7 oct. 2023. URL : <https://ctan.org/pkg/siunitx>.

LES INDEX

INDEX GÉNÉRAL

- beamer, 83
- biber, 45
- biblatex, 45
- bibliographie, 45
- bibtex, 45
- block, 88

- caractères spéciaux, 27
- changement de corps, 29
- classe de document, 21
- composition de document, 8
- couches, 89
- courbes de Bézier, 99

- dessin vectoriel, 94
- diapositive, 85
- distribution, 13

- entrée bibliographique, 46
- environnement, 30

- extension, 22

- flottants, 43

- gestion de bibliographie, 48

- images, 37
- inclure des images, 37

- latexmk, 19
- liste des flottants, 44
- listes, 33
- logiciel libre, 11
- Lua \LaTeX , 19

- MathSciNet, 48
- mathématiques
 - accents, 69
 - délimiteurs+, 74
 - espaces, 66
 - flèches, 69

fonctions et opérateurs, 76
 indices et exposants, 65
 lettres grecques, 67
 matrices, 77
 modes de composition, 63
 opérateurs binaires, 67
 relations binaires, 69
 styles, 73
 symboles, 75
 texte, 66
 théorèmes, 80
 mot, 24

 OpenType, 19
 overlays, 89

 package, 22, 101
 paragraphe, 25
 PDF, 18
 préambule, 22
 pseudo code, 121

 références croisées, 41

 sectionnement, 25
 style, 27
 Style bibliographique, 51
 symboles mathématiques
 opérateurs binaires, 68
 système international, 104

 tableaux, 35
 T_EXnique, 11
 Tikz, 94
 typographie, 9

 utf8, 19

 WYSIWYG, 8

 zbMATH, 48

 éditeur de texte, 15

 τέχνη, 9

INDEX DES COMMANDES L^AT_EX

`\autocite`, 50
`\autocites`, 50

`\bfseries`, 29
`\bigskip`, 25

`\chapter`, 26
`\chapter*`, 26
`\cite`, 49
`\cites`, 49
`\color`, 102
`\colorbox`, 102
`\complexnum`, 105
 controls, 99
`\coordinate`, 96

`\DeclareMathOperator`, 77
`\declaretheorem`, 110
`\declaretheoremstyle`, 111
`\definecolor`, 101
`\derpart`, 109
`\documentclass`, 21
`\draw`, 97
`\drv`, 109

`\emph`, 29
`\eqref`, 79

`\fcolorbox`, 102
`\fill`, 97
`\footnotesize`, 30

`\frac`, 74
`\framesubtitle`, 85
`\frametitle`, 85
`\Huge`, 30
`\huge`, 30
`\imath`, 71
`\includegraphics`, 37
`\index`, 59
`\intgen`, 109
`\itshape`, 29
`\jmath`, 71
`\label`, 42
`\LARGE`, 30
`\Large`, 30
`\large`, 30
`\left`, 74
`\linewidth`, 38
`\listoftheorems`, 111
`\lstdefinestyle`, 120
`\lstset`, 119
`\mathbb`, 73
`\mathbf`, 73
`\mathcal`, 73
`\mathit`, 73
`\mathrm`, 73
`\mathsf`, 73
`\mathtt`, 73
`\mdseries`, 29
`\medskip`, 25
`\newtheorem`, 80
`\normalsize`, 30
`\num`, 105
`\nuplet`, 109
`\odif`, 110
`\odv`, 110
`\only`, 90
`\overbrace`, 71
`\overleftarrow`, 71
`\overline`, 71
`\overrightarrow`, 71
`\pagecolor`, 102
`\pageref`, 43
`\paragraph`, 26
`\part`, 26
`\pdv`, 110
`\printbibliography`, 51
`\ref`, 42
`\right`, 74
`\rmfamily`, 29
`\scriptsize`, 30
`\scshape`, 29
`\section`, 26
`\sffamily`, 29
`\SI`, 105
`\sisetup`, 105
`\slshape`, 29
`\small`, 30
`\smallskip`, 25
`\sqrt`, 74
`\stackrel`, 72
`\subparagraph`, 26
`\subsection`, 26
`\subsubsection`, 26
`\tableofcontents`, 26, 87
`\text`, 66
`\textbf`, 28
`\textcolor`, 102
`\textit`, 28
`\textmd`, 28
`\textrm`, 28
`\textsc`, 28
`\textsf`, 28
`\textsl`, 28
`\texttt`, 28

<code>\textup</code> , 28	<code>\underbrace</code> , 71
<code>\tiny</code> , 30	<code>\underline</code> , 71
<code>\titlepage</code> , 86	<code>\upshape</code> , 29
<code>\todo</code> , 113	
<code>\ttfamily</code> , 29	<code>\widehat</code> , 71
<code>\uncover</code> , 90	<code>\widetilde</code> , 71

INDEX DES ENVIRONNEMENTS L^AT_EX

alertblock, 88	frame, 85
algorithm, 122	itemize, 106
algorithmic, 122	itemize, 33
align, 79	
	lstlisting, 118
block, 88	math, 63
bmatrix, 77	matrix, 77
center, 31	pmatrix, 77
	quotation, 31
description, 107	quote, 31
description, 34	
	table, 43
enumerate, 106	tabular, 35
enumerate, 34	tblr, 112
equation, 79	tcolorbox, 108
exampleblock, 88	tikzpicture, 94
figure, 44	verbatim, 32
flushleft, 31	
flushright, 31	