

Rédaction avec $\text{\title{L\AA T_{E}X}}$






UNIVERSITÉ
LAVAL

Rédaction avec `\title{LATEX}`

Vincent Goulet

Professeur titulaire
École d'actuariat, Université Laval

©   © 2015-2024 par Vincent Goulet. « Rédaction avec \LaTeX » est mis à disposition sous licence **Attribution-Partage dans les mêmes conditions 4.0 International** de Creative Commons. En vertu de cette licence, vous êtes autorisé à :

- ▶ **partager** — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats;
- ▶ **adapter** — remixer, transformer et créer à partir du matériel pour toute utilisation, y compris commerciale.

L'Offrant ne peut retirer les autorisations concédées par la licence tant que vous appliquez les termes de cette licence.

Selon les conditions suivantes :



Attribution — Vous devez créditer l'œuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'œuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son œuvre.



Partage dans les mêmes conditions — Dans le cas où vous effectuez un remix, que vous transformez, ou créez à partir du matériel composant l'œuvre originale, vous devez diffuser l'œuvre modifiée dans les mêmes conditions, c'est-à-dire avec la même licence avec laquelle l'œuvre originale a été diffusée.

Code source

 [Voir sur Gitlab](#)

Couverture

Suricates (*Suricata suricatta*) en Namibie. Parfois surnommé « sentinelle du désert », ce petit carnivore vit dans le sud-ouest de l'Afrique. Très prolifique, le suricate vit en groupes familiaux au sein d'une colonie. Crédit photo : © Sara&Joachim&Mebe, [CC BY-SA 2.0](#) via [Wikimedia Commons](#).

Concept original du titre : Marie-Ève Guérard.

Introduction


Le présent ouvrage tire son origine d'une formation sur la rédaction de thèses et de mémoires avec \LaTeX développée pour la Bibliothèque de l'Université Laval. La formation aborde les concepts de base pour un nouvel utilisateur : processus d'édition, compilation, visualisation ; séparation du contenu et de l'apparence du texte ; mise en forme du texte ; séparation du document en parties ; rudiments du mode mathématique. Transformée en prose, la série de diapositives qui appuie la présentation correspond grosso modo aux quatre premiers chapitres de l'ouvrage.

Les six autres chapitres visent à rendre l'utilisateur de \LaTeX débutant ou intermédiaire autonome dans la rédaction de documents relativement complexes comportant des tableaux, des figures, des équations mathématiques élaborées, une bibliographie, etc. J'ai aussi émaillé le texte de conseils et d'astuces glanés au fil de mes plus de vingt années d'utilisation du système de mise en page.

Chaque chapitre comporte quelques exercices. Les solutions se trouvent en annexe.

Fonctionnalités interactives

En consultation électronique, ce document se trouve enrichi de plusieurs fonctionnalités interactives.

- ▶ Intraliens entre le numéro d'un exercice et sa solution, et vice versa. Ces intraliens sont aussi marqués par la couleur ■.
- ▶ Intraliens entre les citations dans le texte et leur entrée dans la bibliographie. Ces intraliens sont marqués par la couleur ■.
- ▶ Hyperliens vers des ressources externes marqués par le symbole  et la couleur ■.
- ▶ Table des matières, liste des tableaux et liste des figures permettant d'accéder rapidement à des ressources du document.

- ▶ Index comprenant les références aux commandes et environnements \LaTeX , ainsi qu'aux noms de paquetages et de classes.

Blocs signalétiques

Le document est parsemé de divers types de blocs signalétiques inspirés de [AsciiDoc](#) qui visent à attirer votre attention sur une notion.



Astuce! Ces blocs contiennent un truc, une astuce, ou tout autre type d'information utile.



Important! Ces blocs contiennent les remarques les plus importantes. Veillez à en tenir compte.



Ces blocs contiennent des remarques additionnelles sur la matière ou des informations amusantes, mais non essentielles.



Ces blocs contiennent des renvois vers des vidéos dans la [chaîne YouTube](#) liée ce document.

Liens vers la documentation

■ `memoir`

Parmi les hyperliens vers des ressources externes, le texte compte plusieurs renvois vers la documentation d'un paquetage ou d'une classe, par exemple vers la [documentation](#) de la classe `memoir`. L'hyperlien mène vers la documentation dans le site [TeXdoc Online](#). On trouve également dans la marge le nom du fichier correspondant (sans l'extension `.pdf`) sur un système doté de \TeX Live.




La plupart des logiciels intégrés de rédaction \LaTeX offrent une interface pour accéder à cette documentation.

- ▶ TeXShop : menu Aide|Afficher l'aide pour le package ($\text{\C} \text{\#} I$).
- ▶ Texmaker : menu Aide|TeXDoc [selection].
- ▶ GNU Emacs : commande `TeX-doc` (`C-c ?`) du mode spécialisé AUCTeX .

Le lecteur devrait consulter la rubrique d'aide de son éditeur pour savoir s'il offre une interface au système de gestion de la documentation `Texdoc` de \TeX Live.

Références additionnelles



L'ouvrage n'a aucune prétention d'exhaustivité. La consultation de documentation additionnelle pourra s'avérer nécessaire pour réaliser des mises en page plus élaborées. À cet égard, je recommande chaudement le livre de [Kopka et Daly \(2003\)](#) — il a servi d'inspiration pour ce document à maints endroits. La très complète documentation (plus de 600 pages !) de la classe memoir ([Wilson, 2022](#)) constitue une autre référence de choix. Je recommande également :

- ▶ [LaTeX dans Wikilivre](#)  pour de la documentation en ligne, en français et libre ;
- ▶ le très actif forum de discussion [TeX-~~La~~TeX Stack Exchange](#)  (avant de penser y poser une question, vérifiez que la réponse ne se trouve pas déjà dans le forum... ce qui risque fort d'être le cas) ;
- ▶ la vénérable et très complète foire aux questions [Frequently Asked Questions List for TeX](#)  (en anglais).


Installation d'une distribution

L'utilisation de \LaTeX requiert évidemment une distribution du système. Je recommande la distribution TeX Live administrée par le TeX Users Group.




Visionnez les vidéos qui expliquent comment installer la distribution TeX Live [sur macOS](#)  et [sur Windows](#) .

Fichiers d'accompagnement

Cet ouvrage devrait être accompagné des fichiers nécessaires pour compléter certains exercices figurant à la fin des chapitres, ainsi que d'un gabarit `exercice-gabarit.tex` pour composer les solutions des autres exercices. Si ce n'est pas le cas, récupérez les fichiers dans le site [Comprehensive TeX Archive Network \(CTAN\)](#) .

Document libre

Tout comme TeX, \LaTeX et l'ensemble des outils présentés dans ce document, le projet « Rédaction avec \LaTeX » s'inscrit dans le mouvement de [l'informatique libre](#) . Vous pouvez accéder à l'ensemble du code source

en suivant le lien dans la page de copyright. Vous trouverez dans le fichier README.md toutes les informations utiles pour composer le document.

Table des matières

Introduction **vii**

Table des matières **xi**

Liste des tableaux **xv**

Liste des figures **xvii**

1	Présentation de T_EX et L^AT_EX	1
1.1	Description sommaire	2
1.2	Quelques choses simples à réaliser avec L ^A T _E X	3
1.3	Processus de création d'un document	4
1.4	Outils de production	4
1.5	Exercices	6
2	Principes de base	7
2.1	Séparation du contenu et de l'apparence	7
2.2	Règles de saisie	8
2.3	Structure d'un fichier	9
2.4	Classes et paquetages	10
2.5	Commandes	11
2.6	Environnements	13
2.7	Longueurs	14
2.8	Commentaires	15
2.9	Caractères spéciaux	15
2.10	L ^A T _E X en français	19
2.11	Exercices	21
3	Organisation d'un document	23
3.1	Choix d'une classe	23

3.2	Parties d'un document	27
3.3	Table des matières	31
3.4	Renvois automatiques	33
3.5	Document contenu dans plusieurs fichiers	37
3.6	Exercices	40
4	Apparence et disposition du texte	43
4.1	Police de caractères	43
4.2	Taille du texte	44
4.3	Italique	45
4.4	Listes	46
4.5	Texte centré	48
4.6	Citations	48
4.7	Notes de bas de page	49
4.8	Texte brut et code source	50
4.9	Exercice	51
5	Boîtes	53
5.1	Boîtes horizontales	53
5.2	Boîtes verticales	55
5.3	Boîtes de réglure	57
5.4	Exercices	58
6	Tableaux et figures	61
6.1	De la conception de beaux tableaux	61
6.2	Tableaux	63
6.3	Figures et graphiques	67
6.4	Éléments flottants	73
6.5	Exercices	78
7	Mathématiques	81
7.1	Principes de base du mode mathématique	81
7.2	Un paquetage incontournable	83
7.3	Principaux éléments du mode mathématique	83
7.4	Symboles mathématiques	90
7.5	Équations sur plusieurs lignes et numérotation	95
7.6	Délimiteurs de taille variable	98
7.7	Caractères gras en mathématiques	102
7.8	Exercices	105

8	Bibliographie et citations	107
8.1	Quel système utiliser?	107
8.2	Processus de création d'une bibliographie	109
8.3	Création d'une base de données	109
8.4	Style des citations et de la bibliographie	113
8.5	Insertion de références dans le texte	114
8.6	Composition de la bibliographie	116
8.7	Exercices	120
9	Commandes et environnements définis par l'utilisateur	123
9.1	Nouvelles commandes	124
9.2	Nouveaux environnements	126
9.3	Exercices	127
10	Trucs et astuces divers	129
10.1	Contrôle de la disposition du texte	129
10.2	Au-delà de la police Computer Modern	132
10.3	Couleurs	132
10.4	Hyperliens et métadonnées de documents PDF	134
10.5	Présentation de code informatique	135
10.6	Production de rapports avec l'analyse intégrée	137
10.7	Diapositives	139
10.8	Gestion des versions et travail collaboratif	140
10.9	Exercices	141
A	Classe pour les thèses et mémoires de l'Université Laval	143
B	Solutions des exercices	145
	Chapitre 2	145
	Chapitre 5	145
	Chapitre 6	146
	Chapitre 7	148
	Chapitre 8	149
	Chapitre 9	150
	Chapitre 10	151
	Bibliographie	153
	Index	157

Liste des tableaux

- 1.1 Moteurs et formats les plus courants 5
- 2.1 Principales unités de mesure pour les longueurs dans \LaTeX 15
- 2.2 Principaux enjeux de rédaction en français et paquetages offrant des solutions 21
- 3.1 Caractéristiques des principales classes standards 24
- 3.2 Commandes d'identification de la structure logique d'un livre et leurs effets 31
- 4.1 Commandes de changement d'attribut de la police 44
- 4.2 Commandes de changement de la taille des caractères 45
- 6.1 Tableau correspondant au code de la figure 6.2 76
- 7.1 Points de suspension 87
- 7.2 Espaces dans le mode mathématique 89
- 7.3 Lettres grecques minuscules 92
- 7.4 Lettres grecques majuscules 92
- 7.5 Quelques opérateurs binaires 93
- 7.6 Quelques relations et leur négation 93
- 7.7 Quelques flèches et leur négation 94
- 7.8 Accents et symboles mathématiques divers 94
- 7.9 Comparaison des environnements pour les équations hors paragraphe de **amsmath** 96
- 7.10 Tailles des délimiteurs mathématiques 101
- 8.1 Quelques styles bibliographiques et leur effet sur le mode de citation et le format des notices bibliographiques 114
- 10.1 Couleurs toujours disponibles lorsque le paquetage **xcolor** est chargé 133

Liste des figures

- 2.1 Fichier source \LaTeX simple comportant les deux parties obligatoires : le préambule et le corps du document 10

- 3.1 Exemple de mise en page avec la classe article 24
- 3.2 Exemple de mise en page avec la classe report 25
- 3.3 Exemple de mise en page avec la classe book 25
- 3.4 Illustration d'un titre d'article standard. 29
- 3.5 Consultation d'un document PDF avec la visionneuse Aperçu de macOS 32
- 3.6 Texte produit par le code de l'exemple 3.2 illustrant un renvoi automatique standard 35
- 3.7 Texte produit par le code de l'exemple 3.2 après l'ajout du paquetage **hyperref** 36
- 3.8 Texte produit par le code de l'exemple 3.4 37
- 3.9 Structure type d'un fichier maitre 39

- 5.1 Pour \TeX , un texte n'est qu'un alignement de boites 54

- 6.1 Commandes pour modifier les étiquettes des tableaux et des figures utilisées par **babel** 75
- 6.2 Code source pour créer le tableau 6.1 76
- 6.3 Exemple de disposition de plusieurs graphiques dans une même figure flottante 77


- 8.1 Code source d'un fichier de base de données et d'un document simple 118
- 8.2 Zone de texte du document aux diverses étapes de la compilation des fichiers de la figure 8.1 avec X_{\LaTeX} et $\text{BIB}\TeX$ 119
- 8.3 Interfaces de sélection du programme $\text{BIB}\TeX$ dans TeXShop et Texmaker 120

1 Présentation de $\text{T}_{\text{E}}\text{X}$ et $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

La production d'un document à l'aide d'un ordinateur comporte toujours trois grandes étapes :

1. la saisie du texte dans l'ordinateur ;
2. la composition du texte sur la page en lignes et en paragraphes, ainsi que la disposition des éléments tels que les figures et les tableaux ;
3. la production du document final pour consultation à l'écran ou pour l'impression.

Les logiciels de traitement de texte combinent les trois étapes de manière transparente pour l'utilisateur.

Le système $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, lui, est un logiciel de mise en page qui se charge uniquement de la seconde étape de production, celle qui était autrefois réalisée à la main ou à l'aide d'une machine comme la **Linotype**  par les typographes.

La préparation d'un document avec $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ requiert d'utiliser trois outils bien différents, soit un pour chaque étape de production. Nous verrons rapidement que ce qui peut à priori sembler un inconvénient constitue en fait un grand avantage.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est l'un des systèmes de mise en page les plus utilisés dans le monde, particulièrement dans la production de documents scientifiques intégrant de multiples équations mathématiques, des graphiques, du code informatique, etc. En fait, le nom $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ sert généralement de terme générique pour toute une famille de programmes. Ce chapitre explique donc non seulement la philosophie derrière $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, mais également le lien qui unit ce programme aux autres dont on croise couramment le nom aujourd'hui : $\text{T}_{\text{E}}\text{X}$, $\text{pdfL}^{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{X}_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

Un peu de phonétique avant d'aller plus loin. La racine $\text{T}_{\text{E}}\text{X}$ est une transcription en majuscules des lettres grecques $\tau\epsilon\chi$. La dernière lettre des acronymes ci-dessus n'est donc non pas un X latin, mais plutôt la lettre grecque khi majuscule, qui est visuellement identique. Pour cette raison, la terminai-

son des acronymes se prononce « tek ». Voilà qui devrait déjà vous permettre de distinguer le système de mise en page du matériau élastique.



L'histoire courte *Paul apprenti typographe* publiée dans l'album *Paul à la campagne* (Rabagliati, 1999) explique formidablement bien le processus de composition d'un texte avec la Linotype. Quelques séquences du très recommandable film *The Post* (2017) illustrent également la composition d'une page de journal.

1.1 Description sommaire

À la base de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ et de ses dérivés, il y a toujours le système $\text{T}_{\text{E}}\text{X}$ développé par Donald Knuth à partir de la fin des années 1970 alors qu'il travaillait à la rédaction de son œuvre phare *The Art of Computer Programming*. Comme il n'était pas satisfait de la qualité typographique des systèmes de mise en page alors disponibles, il a tout naturellement décidé d'en créer un à la hauteur de ses exigences !

$\text{T}_{\text{E}}\text{X}$ est un système de mise en page (*typesetting*) ou de préparation de documents. De manière plus technique, $\text{T}_{\text{E}}\text{X}$ est un langage de programmation spécialisé dans la disposition de caractères sur une page. Il met l'accent sur la production de documents de grande qualité à la typographie soignée, surtout pour les mathématiques.

Comme le système et le langage $\text{T}_{\text{E}}\text{X}$ ne sont pas des plus faciles à apprendre, Leslie Lamport a créé en 1983 un ensemble de macro-commandes baptisé $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ pour faciliter l'utilisation de $\text{T}_{\text{E}}\text{X}$. Le terme en est venu, chez les utilisateurs, à nommer l'ensemble du système.

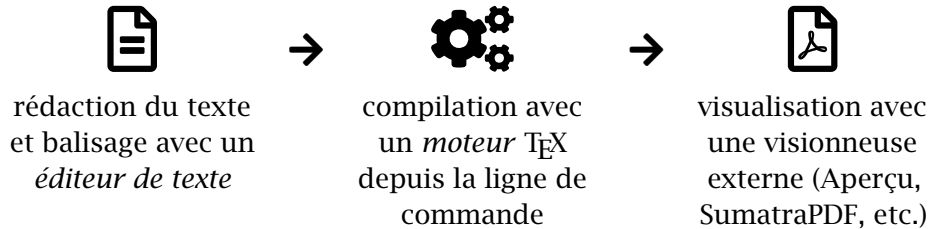
$\text{T}_{\text{E}}\text{X}$ et $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ sont des langages de balisage (*Markup Languages*) qui indiquent la mise en forme du texte à l'intérieur de celui-ci par le biais de commandes spéciales.

Exemple 1.1. Les traitements de texte sont d'abord et avant tout conçus pour respecter le plus fidèlement possible ce que l'utilisateur produit (d'où l'appellation *What You See is What You Get*, WYSIWYG). Les systèmes de mise en page, quant à eux, visent à maximiser la qualité de la mise en page et de la typographie pour un texte donné.






Voici deux exemples de typographie soignée. D'abord, l'utilisation de *ligatures* (jonctions) entre certaines lettres. À gauche, ce que produisent les traitements de texte, qui ne voient qu'une série de lettres individuelles. À droite, le résultat avec $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, qui peut analyser le texte et identifier les ligatures.

1.3 Processus de création d'un document



Le processus de création d'un document $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ compte trois phases : la rédaction, la compilation (ou composition par le système) et la visualisation du résultat. On peut représenter ce processus schématiquement ainsi :



Les logiciels de rédaction intégrés facilitent grandement les deux premières étapes — certains intègrent même une visionneuse PDF pour englober le processus au complet. Il existe plusieurs de ces logiciels. Mentionnons, par exemple :

- ▶ [Texmaker](#)  (multiplateforme) ;
- ▶ [TeXworks](#)  (multiplateforme) ;
- ▶ [TeXShop](#)  (macOS seulement) ;
- ▶ [WinEdt](#)  (Windows seulement) ;
- ▶ [GNU Emacs](#)  (multiplateforme) et à peu près tous les bons éditeurs de texte pour programmeur.



Visionnez les vidéos qui illustrent comme composer un document simple de type *Hello, World!* avec [TeXShop sur macOS](#)  et avec [Texmaker sur Windows](#) .


1.4 Outils de production



Un petit mot d'abord sur les arcanes de $\text{T}_{\text{E}}\text{X}$. Dans ce monde, il existe des *moteurs* et des *formats*. Un moteur $\text{T}_{\text{E}}\text{X}$ est un programme informatique qui transforme du code source en représentation d'un document sur une page. Un format est simplement un ensemble de macro-commandes comprises par un moteur et qui est chargé par défaut lorsque le moteur est invoqué avec un nom de commande spécifique.

Le [tableau 1.1](#) dresse la liste des divers *moteurs* $\text{T}_{\text{E}}\text{X}$ et des formats (ensembles de macro-commandes) couramment utilisés aujourd'hui.


Moteur	Format	Fichier de sortie
<code>tex</code>	<i>plain</i> T _E X	DVI
<code>tex (latex)</code>	L ^A T _E X	DVI
<code>pdftex (pdf_latex)</code>	pdfL ^A T _E X	PDF
<code>xetex (xelatex)</code>	X _Y L ^A T _E X	PDF
<code>luatex (lua_latex)</code>	LuaL ^A T _E X	PDF

TAB. 1.1 – Moteurs et formats les plus courants

- ▶ Les formats les plus usuels sont pdfL^AT_EX et X_YL^AT_EX.
- ▶ Le moteur `pdftex` est le moteur par défaut des distributions L^AT_EX modernes. Comme son nom l'indique, ce moteur produit directement un fichier de sortie en format PDF. C'est la principale différence par rapport au moteur `tex`.
- ▶ Le moteur `xetex` peut utiliser directement les polices de caractères du système d'exploitation. La [section 10.2](#) traite plus en détail de ce sujet.
- ▶ Le moteur `luatex` et le format LuaL^AT_EX offrent les mêmes avantages que `xetex` et X_YL^AT_EX en plus d'intégrer les fonctionnalités du langage de programmation [Lua](#) .
- ▶ Le format de fichier de sortie DVI, qui est antérieur aux formats PostScript et PDF, permet de décrire la disposition d'un document exactement telle qu'elle devrait apparaître à l'écran ou à l'impression. C'est un format aujourd'hui plus ou moins tombé en désuétude depuis la standardisation autour du format PDF.

Le système L^AT_EX est formé d'un grand nombre de composantes réunies sous forme d'une *distribution*. La plus populaire distribution aujourd'hui est [T_EX Live](#)  — ou sa variante [MacT_EX](#)  pour macOS. Elle est administrée par le T_EX Users Group. La philosophie de T_EX Live : tout installer. Cette façon de faire est aujourd'hui réalisable puisque l'espace disque est disponible à profusion dans les ordinateurs. C'est également la plus simple puisque à peu près tout ce que l'on est susceptible d'utiliser dans un système T_EX est déjà installé.

L'[introduction](#) fournit des liens vers des vidéos expliquant comment installer la distribution T_EX Live sur macOS et Windows.

La distribution [MiK_TE_X](#)  demeure également très populaire sur Windows.



Quelques faits amusants au sujet de T_EX.

- ▶ T_EX est aujourd'hui considéré exempt de bogue. D'ailleurs, Donald Knuth vous offre une récompense (symbolique) si vous en trouvez un!
- ▶ Le numéro de version de T_EX converge vers π . Au moment d'écrire ces lignes, le numéro de version était 3.141592653.

Consultez les liens suivants pour en savoir plus.

- ▶ [Histoire de T_EX](#) sur le site du T_EX Users Group (en anglais).
- ▶ T_EX sur Wikipedia ([version française](#); [version anglaise](#), plus complète).
- ▶ [Texte de la conférence](#) de Beebe (2004) à l'occasion des 25 ans de T_EX et METAFONT.

1.5 Exercices

- 1.1 Composer un premier document très simple de type *Hello World!* tel qu'expliqué dans les vidéos mentionnées à la [section 1.3](#).
- 1.2 À l'aide d'un logiciel intégré de rédaction (Texmaker et TeXShop¹ constituent des bonnes options pour débuter), ouvrir et compiler le fichier `exercice-minimal.tex`.
- 1.3 Question de voir ce que L^AT_EX peut faire, compiler le document élaboré `exercice-demo.tex` de la manière suivante : une fois avec L^AT_EX; une fois avec BIB_TE_X; deux à trois autres fois avec L^AT_EX.

1. Avant d'ouvrir un premier fichier avec TeXShop, ouvrez les préférences de l'application et changez l'encodage par défaut pour « Unicode (UTF-8) ». Redémarrez ensuite l'application.

2 Principes de base

Un système de mise en page tel que \LaTeX repose sur une logique de séparation entre l'apparence d'un document et sa structure. Si vous avez l'habitude d'utiliser un traitement de texte, vous devrez fort probablement vous défaire d'une vilaine habitude : vous préoccuper sans cesse, au moment de la rédaction, de la disposition du texte.

Ce principe accepté, il vous faudra néanmoins indiquer au logiciel la structure du document. Avec \LaTeX cela s'effectue par le biais de diverses instructions que l'on insère au fil du texte. À la base, les logiciels de traitements de texte n'opèrent pas différemment, sauf qu'ils cachent les codes aux utilisateurs¹. Sous prétexte de simplicité d'utilisation, ils causent en fait bien des maux de tête. Qui ne s'est pas déjà demandé, en utilisant un traitement de texte moderne : « Pourquoi donc mon texte est-il soudainement en gras ? »

Ce chapitre explique comment aborder la rédaction d'un document avec \LaTeX ainsi que la syntaxe de base des différents types d'instructions que l'on peut insérer dans un texte pour en spécifier la structure et la mise en forme.

2.1 Séparation du contenu et de l'apparence

Lors de la rédaction avec un système de mise en page tel que \LaTeX , on se concentre sur le contenu et la *structure* du document, et non pas sur son *apparence*. Par exemple :

- au lieu de prescrire qu'un titre de section doit être en gras 14 points, on indique simplement à \LaTeX que le texte doit être traité comme un titre de section ;

1. Le leader du traitement de texte jusqu'au milieu des années 1990, Word Perfect, offrait l'option d'afficher l'ensemble des codes de mise en page. C'est malheureusement une caractéristique brillante que Microsoft Word et les autres logiciels développés depuis ont choisi d'omettre.

`\textbf{\large Titre}` → `\section{Titre}`

- ▶ au lieu de décider qu'un mot sur lequel l'on souhaite insister sera en italique, on indique à \LaTeX de mettre de l'emphase sur ce mot sans se soucier de la mise en forme.

`\textit{texte}` → `\emph{texte}`

L'apparence du texte sera prise en charge par \LaTeX . Comme les gabarits sont l'œuvre de spécialistes en typographie, il est généralement préférable de ne pas les modifier. À titre d'exemple, \LaTeX détermine automatiquement la largeur des marges en fonction de la taille de la police de manière à ce que les lignes de texte comptent approximativement 70 caractères. La raison : lorsqu'une ligne de texte est trop longue, notre œil a de la difficulté à la suivre sur toute sa longueur. Il a tendance à passer à la ligne inférieure, ce qui rend la lecture plus difficile.

2.2 Règles de saisie

Une fois le principe de séparation du contenu et de l'apparence compris et accepté, veillez à respecter les règles simples suivantes lors de la saisie du texte.

1. On sépare les mots par une ou plusieurs *espaces*. Qu'il y en ait une ou un millier, seule la première compte et la mise en page sera la même.

Les espaces délimitent les mots. Leur nombre n'a pas d'importance.

Les espaces délimitent les mots. Leur nombre n'a pas d'importance.

Les espaces délimitent les mots. Leur nombre n'a pas d'importance.

Les espaces délimitent les mots. Leur nombre n'a pas d'importance.

2. On sépare les paragraphes par une ou plusieurs lignes blanches. Celles-ci n'apparaîtront pas nécessairement dans le texte final; les gabarits standards identifient les paragraphes par un retrait de première ligne.

Les lignes blanches délimitent les paragraphes.

Une ou plusieurs, ça ne fait aucune différence!

Les lignes blanches délimitent les paragraphes.

Une ou plusieurs, ça ne fait aucune différence!

Les lignes blanches délimitent les paragraphes.

Une ou plusieurs, ça ne fait aucune différence!

Les lignes blanches délimitent les paragraphes.

Une ou plusieurs, ça ne fait aucune différence!

3. On utilise des *commandes* pour indiquer la structure du texte dans le texte. Celles-ci débutent presque toujours par le symbole « \ ». À la différence des logiciels de traitement de texte, les instructions de mise en forme du document sont donc toujours visibles et, par conséquent, modifiables facilement et sans surprise (on ne se demande donc jamais où se termine le gras).

Les commandes sont visibles dans le `\textbf{texte}`, mais évidemment pas dans le `\emph{document}` fini.

Les commandes sont visibles dans le **texte**, mais évidemment pas dans la *document* fini.

2.3 Structure d'un fichier

Un fichier source \LaTeX — dont vous trouverez un exemple simple à la [figure 2.1](#) — est toujours composé de deux parties : le préambule et le corps du document.

Préambule Suite de commandes spécifiant la mise en forme globale du document (format du papier, marges, entête et pied de page, etc.). Il contient au minimum la commande `\documentclass`. Les commandes contenues dans le préambule ont un effet global sur le document. Les lignes 1-5 forment le préambule dans l'exemple de la [figure 2.1](#).

Corps du document Contenu du document en tant que tel. Il débute par `\begin{document}` et se termine par `\end{document}`. Le corps du

```

1 \documentclass[11pt, french]{article}
2   \usepackage{babel}
3   \usepackage[autolanguage]{numprint}
4   \usepackage[utf8]{inputenc}
5   \usepackage[T1]{fontenc}
6
7 \begin{document}
8
9 Lorem ipsum dolor sit amet, consectetur
10 adipiscing elit. Donec quam nulla, bibendum
11 vitae ipsum vel, fermentum pellentesque
12 orci.
13
14 \end{document}

```

FIG. 2.1 – Fichier source \LaTeX simple comportant les deux parties obligatoires : le préambule (lignes 1-5) et le corps du document (lignes 7-14).

document peut aussi contenir des commandes, mais l'effet de celles-ci demeure presque toujours local. Les lignes 7-14 du code de la [figure 2.1](#) forment le corps du document.

2.4 Classes et paquets

La première commande du préambule est normalement la déclaration de la *classe* du document. La forme de la déclaration est la suivante :

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

Les classes standards de \LaTeX sont article, report, book, letter et slides. La [section 3.1](#) traite des différences entre les diverses classes et des *options* disponibles.

Les *paquetages* permettent de modifier des commandes ou d'ajouter des fonctionnalités à \LaTeX . On charge les paquetages dans le préambule avec des commandes de la forme

```

\usepackage{paquetage}
\usepackage[options]{paquetage}
\usepackage{paquetage1,paquetage2, ...}

```

La première et la troisième forme permettent de charger un ou plusieurs paquetages sans options. La seconde permet de spécifier des *options* au chargement du paquetage. Il n'est évidemment pas possible de préciser des options avec la troisième forme puisque \LaTeX ne saurait à quel paquetage celles-ci se rapportent.

Certains paquetages permettent que leurs options apparaissent parmi les *options* de la commande `\documentclass`. Elles sont ainsi plus « visibles » pour d'autres paquetages. Par exemple, l'option `french` que l'on retrouve dans la déclaration de la classe à la [figure 2.1](#) est en fait une option du paquetage `babel`.

2.5 Commandes

J'ai déjà fait référence à quelques reprises au concept de commande \LaTeX . Cette section se penche sur leur syntaxe.

Les formes générales des commandes \LaTeX sont :

```
\<nomcommande>[<arg_optionnel>]{<arg_obligatoire>}
\<nomcommande>*[<arg_optionnel>]{<arg_obligatoire>}
```

Ici, *<nomcommande>* est le nom de la commande. Il débute par le symbole « \ » et il est exclusivement formé de lettres, habituellement des minuscules (\LaTeX est sensible à la casse). La forme étoilée d'une commande réalise généralement une action légèrement différente de la version sans étoile. Par exemple, la commande `\section` crée une nouvelle section numérotée, alors que `\section*` n'insère aucune numérotation.

Lorsque la commande accepte des arguments, les arguments obligatoires sont placés entre accolades `{ }` et les arguments optionnels sont placés entre crochets `[]`.

Certaines commandes n'ont aucun argument. Leur forme est alors

```
\<nomcommande>
```

Dans ce cas, le nom de la commande se termine par tout symbole qui n'est pas une lettre — y compris l'espace ! Cette règle fait en sorte qu'une espace après le nom d'une commande est considérée comme un marqueur de la fin du nom de la commande. Cette règle joue parfois de vilains tours en « avalant » l'espace entre une commande et le mot qui suit ; voir l'[exemple 2.1](#) et l'[exercice 2.2](#).

La portée d'une commande est limitée à la zone entre accolades `{ }`, le cas échéant.

Exemple 2.1. Voici trois exemples de commandes \LaTeX : une sans argument, une avec un seul argument obligatoire et une commande avec deux arguments obligatoires et un argument optionnel.

1. La commande \LaTeX permet de composer de logo \LaTeX .

Apprendre \LaTeX c'est formidable!	Apprendre \LaTeX c'est formidable!
---	---

Vous pouvez constater ici que l'espace suivant le nom de la commande a été interprétée par \LaTeX comme un marqueur de la fin de la commande et qu'elle a été supprimée du texte. Deux possibilités pour contourner cette particularité du langage : fournir un argument vide à la commande, ou placer celle-ci entre accolades pour limiter sa portée à elle-même :


Apprendre $\text{\LaTeX}\{\}$ c'est formidable!	Apprendre \LaTeX c'est formidable!
---	---

Apprendre $\{\text{\LaTeX}\}$ c'est formidable!	Apprendre \LaTeX c'est formidable!
---	---

2. La commande \emph met de l'emphase (en général sous forme d'italique) sur le ou les mots en argument.

Il est $\text{\emph}\{\text{essentiel}\}$ de connaître la syntaxe de $\{\text{\LaTeX}\}$.	Il est <i>essentiel</i> de connaître la syntaxe de \LaTeX .
--	--

3. La commande \rule produit un rectangle plein. Elle a deux arguments obligatoires : la longueur et la hauteur du rectangle, dans l'ordre. Un argument optionnel permet de surélever le rectangle au-dessus de la ligne de base (voir le [chapitre 5](#) pour plus de détails).

Réglure de 1~cm de long et 3~mm d'épais surélevée de 2~points au-dessus de la ligne de base: $\text{\rule}[2pt]\{1cm\}\{3mm\}$.	Réglure de 1 cm de long et 3 mm d'épais surélevée de 2 points au-dessus de la ligne de base :  .
---	---

□

Exemple 2.2. La commande `\bfseries` sélectionne une police grasse pour tout le texte qui suit.

En typographie, la `\bfseries` graisse est l'épaisseur d'un trait ou d'un caractère.

En typographie, la **graisse est l'épaisseur d'un trait ou d'un caractère.**

Pour limiter le changement à une zone de texte, il faut la délimiter par des accolades.

En typographie, la `{\bfseries graisse est l'épaisseur d'un trait}` ou d'un caractère.

En typographie, la **graisse est l'épaisseur d'un trait** ou d'un caractère.

□

Vous pouvez définir des nouvelles commandes \LaTeX à loisir. Ceci est expliqué au [chapitre 9](#).

2.6 Environnements

Un environnement \LaTeX est une zone de texte délimitée par une construction du type

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

Le contenu d'un environnement est traité différemment du reste du texte en fonction des paramètres de l'environnement. Par exemple, le texte à l'intérieur d'un environnement `center` est centré sur la page.

Les changements induits par un environnement s'appliquent uniquement à l'intérieur de celui-ci. Il en va de même des commandes utilisées à l'intérieur d'un environnement.

Exemple 2.3. L'environnement `quote` sert à composer des citations. Le texte à l'intérieur de l'environnement sera placé dans un bloc séparé du texte principal et en retrait des marges gauche et droite.

```
La phrase
\begin{quote}
  Attention aux bogues dans
  le code ci-dessus; je ne
  l'ai pas testé, j'ai
  seulement prouvé qu'il
  était correct.
\end{quote}
est une citation célèbre du
créateur de {\TeX}, Donald
Knuth.
```

La phrase

Attention aux bogues dans le code ci-dessus; je ne l'ai pas testé, j'ai seulement prouvé qu'il était correct.

est une citation célèbre du créateur de \TeX , Donald Knuth.

Si la citation est dans la langue originale, il est préférable de la composer en italique.

```
La phrase
\begin{quote}
  \itshape
  Beware of bugs in the above
  code; I have only proved it
  correct, not tried it.
\end{quote}
est une citation célèbre du
créateur de {\TeX}, Donald
Knuth.
```

La phrase

Beware of bugs in the above code; I have only proved it correct, not tried it.

est une citation célèbre du créateur de \TeX , Donald Knuth.

On remarque que l'effet de la commande `\itshape` s'est limité à l'environnement. □

2.7 Longueurs

Plusieurs commandes \TeX requièrent en argument une mesure de largeur ou de hauteur. Dans la terminologie de \TeX , on parle plus généralement de longueur (*length*).

Une longueur est un nombre positif, négatif ou nul *obligatoirement* et *immédiatement* suivi d'un symbole d'unité de mesure. Le [tableau 2.1](#) présente les principales unités de mesure utilisées par \TeX et le symbole correspondant.

TAB. 2.1 – Principales unités de mesure pour les longueurs dans \LaTeX

Nom ou description	Symbole	Longueur équivalente
millimètre	mm	
centimètre	cm	10 mm
pouce	in	2,54 cm
point	pt	1/72,27 pouce
point PostScript	bp	1/72 pouce
largeur de la lettre M	em	fonction de la police
hauteur de la lettre x	ex	fonction de la police

Il existe un certain nombre de longueurs prédéfinies. Les plus utiles sont `\linewidth`, qui contient la largeur de la ligne de texte courante, et `\textwidth`, qui contient la largeur de la page courante. Dans du texte normal, les deux mesures sont habituellement égales.

2.8 Commentaires

Le symbole «%» indique un commentaire dans le code source : tout le texte après le symbole jusqu'à la fin de la ligne est ignoré par \LaTeX .

```
| texte % ignoré par LaTeX | texte
```

2.9 Caractères spéciaux

Les claviers d'ordinateur mettent à la disposition des auteurs toutes les lettres de l'alphabet (en versions minuscule et majuscule), les chiffres de 0 à 9, un certain nombre de symboles et, selon le clavier, des versions accentuées de certaines lettres. L'entrée des lettres et des chiffres ne pose pas de problème particulier pour \LaTeX , mais certains symboles sont réservés. De plus, certains symboles d'usage courant ne sont pas disponibles sur les claviers.

2.9.1 Espaces et retours à la ligne

J'ai déjà abordé à la [section 2.2](#) le traitement spécial réservé par \LaTeX aux espaces et aux retours à la ligne dans le code source. Les précisions suivantes s'imposent :

- ▶ seule la première espace entre deux éléments compte ;

- ▶ les espaces en début de ligne sont ignorées ;
- ▶ un retour à la ligne simple est traité comme une espace ;
- ▶ il faut deux retours à la ligne consécutifs (ce qui résulte en une ligne blanche dans le code source) pour identifier un changement de paragraphe.

Pour forcer une espace à un endroit où \LaTeX la supprimerait normalement, utiliser la commande $\backslash_$ (le symbole « \backslash » suivi d'une espace, représentée ici par le symbole $_$).

```
Apprendre \LaTeX\ c'est
formidable!
```

Apprendre \LaTeX c'est formidable!

Dans le même ordre d'idées, le symbole « \sim » insère une espace insécable entre deux mots, de telle sorte que \LaTeX ne pourra placer les mots sur des lignes différentes. Vous devriez insérer ce symbole dans les noms ou entre une quantité et son unité.

```
M.~Tremblay me doit au
moins 200~\$.
```

M. Tremblay me doit au moins
200 \$.

Il peut arriver que l'espace générée par un retour à la ligne *simple* s'avère indésirable. Dans de tels cas, placez un symbole de commentaire « $\%$ » à la fin de la ligne.

```
Donald Knuth est un dieu
\textsuperscript{[citation]}
.
```

Donald Knuth est un dieu^[citation] .

```
Donald Knuth est un dieu%
\textsuperscript{[citation]}%
.
```

Donald Knuth est un dieu^[citation].

2.9.2 Caractères réservés

Comme à peu près tous les langages de programmation, \TeX réserve certains caractères pour son usage interne. Les caractères suivants sont interprétés comme des commandes :

\$ & ~ _ ^ % { }

Pour utiliser les symboles ci-dessus tels quels dans le texte, il faut les précéder par le symbole « \ » :

\# #	\\$ \$	\& &	_ _
\% %	\{ {	\} }	

(Les commandes \^ et \~ servent à créer des accents ; voir la [section 2.9.5](#).)

2.9.3 Guillemets

L'usage des guillemets nécessite une attention particulière dans \LaTeX . Vous ne devriez pas utiliser dans le code source les guillemets doubles « " » qui se trouvent couramment sur les claviers d'ordinateurs.

Pour obtenir les guillemets anglais, il faut utiliser deux symboles d'accent grave côte à côte pour les guillemets ouvrants « “ » et deux apostrophes côte à côte pour les guillemets fermants « ” ».

``guillemets anglais''	“guillemets anglais”
------------------------	----------------------


En typographie française, il convient d'utiliser les chevrons (« ») comme guillemets. Avec la configuration appropriée de **babel** (voir la [section 2.10](#)), vous pourrez utiliser directement dans le code source les symboles « et » disponibles sur un clavier français. L'espacement requis autour des symboles est géré automatiquement.

«guillemets français»	« guillemets français »
-----------------------	-------------------------

« guillemets français »	« guillemets français »
-------------------------	-------------------------

En français, les guillemets anglais ne servent que pour les citations incluses, c'est-à-dire pour les citations à l'intérieur d'une citation.

Elle m'a dit: «Laurent fait dire ``non merci''.»	Elle m'a dit : « Laurent fait dire “non merci”. »
--	---

Les bons éditeurs de texte adaptés pour \LaTeX redéfinissent l'action de la touche  du clavier pour insérer les symboles appropriés selon la langue du document.

Le paquetage `csquotes` (Lehman et Wright, 2022) propose une autre approche pour la saisie des guillemets. Il fournit une commande `\enquote` qui entoure son argument des guillemets appropriés selon le contexte et la langue du texte. Cette solution peut s'avérer intéressante pour les nouveaux venus à \LaTeX qui n'ont pas encore pris l'habitude d'entrer directement les bons guillemets.

2.9.4 Trait d'union et tirets

On trouve en typographie soignée trois sortes de tirets de longueurs différentes :

1. le trait d'union « - » qui sert à relier des mots entre eux;
2. le tiret demi-cadratin « - » qui sert à joindre deux éléments qui comportent déjà des traits d'union (*Trois-Rivières-Québec*);
3. le tiret cadratin « — » qui sert à identifier le changement d'interlocuteur dans les dialogues ou qui remplacent parfois les parenthèses dans du texte normal.

S'il est simple d'entrer un trait d'union à l'ordinateur, il est rare que les claviers comportent des touches pour entrer les tirets demi-cadratin et cadratin (ou alors cela se fait via une obscure combinaison de touches). \LaTeX permet de créer facilement tous les tirets ci-dessus en répétant simplement le trait d'union deux ou trois fois.

| - - | -- - | --- —

2.9.5 Accents et ligatures

À la base, \LaTeX ne reconnaît pas les lettres accentuées ni les ligatures comme \ae et \oe qui ne sont pas courantes en anglais. Néanmoins, il est possible de créer ces symboles avec des commandes.

| \'{o} ó | \o{} ò | \^{}{o} ô | \"{}{o} ö
 | {\ae} æ | {\AE} Æ | {\oe} œ | {\OE} Œ

Évidemment, entrer tous les accents d'un texte en français avec les commandes ci-dessus se révélerait un véritable cauchemar. Heureusement, avec

la configuration appropriée, il est possible d'entrer directement les lettres accentuées que l'on trouve couramment sur un clavier, voire même certaines ligatures si l'on utilise l'encodage UTF-8. La [section 2.10](#) fournit les détails.

Les commandes pour les accents demeurent utiles pour composer les lettres accentuées des langues européennes autres que le français. Vous trouverez la liste de ces commandes, ainsi que celles pour composer une multitude d'autres symboles, à la section 2 de la très utile *Comprehensive \LaTeX Symbol List* [↗](#) (Pakin, 2021).

comprehensive

2.10 \LaTeX en français

Historiquement, \LaTeX était prévu pour la rédaction de documents en anglais. Fort heureusement, il est aujourd'hui beaucoup plus aisé de rédiger des documents dans d'autres langues.

2.10.1 Approche moderne recommandée

La manière de loin la plus simple, et celle que je recommande vivement, pour rédiger des documents en français consiste à enregistrer le code source dans le codage de caractères UTF-8 [↗](#) et à compiler avec X_{\LaTeX} . Vous pourrez ainsi entrer les lettres accentuées directement au clavier (« é ») sans devoir passer par les commandes \TeX de la section précédente (« \ ' e »).

Vous devrez aussi charger le paquetage **fontspec** (Robertson et Hosny, 2022) dans le préambule pour obtenir les lettres accentuées dans le fichier PDF. Cela ne représente toutefois pas véritablement une étape de configuration additionnelle puisque ce paquetage est fréquemment employé avec X_{\LaTeX} pour contrôler le chargement des polices de caractères (plus de détails à la [section 10.2](#)).

2.10.2 Approche traditionnelle

L'approche plus traditionnelle, basée sur $\text{pdf}\LaTeX$, de rédaction de documents dans une langue autre que l'anglais nécessite le chargement de divers paquetages.


Tout d'abord, il faut charger le paquetage **inputenc** (Jeffrey et Mittelbach, 2021) pour faire en sorte que les lettres accentuées seront reconnues par \LaTeX . Il faut spécifier le type de codage utilisé. Tel que mentionné ci-dessus, vous devriez aujourd'hui utiliser l'UTF-8.

```
\usepackage[utf8]{inputenc}
```

Après avoir configuré l'entrée des caractères, il faut également configurer la sortie. Pour que la coupure automatique des mots contenant des lettres accentuées fonctionne correctement avec pdf \LaTeX , il faut activer le codage de sortie des caractères nommé « T1 » à l'aide du paquetage **fontenc** (Mittelbach et collab., 2016).

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

2.10.3 Typographie et mots clés français

Entrer du texte en français dans le code source n'est pas tout. Il faut adapter \LaTeX au français, qu'il s'agisse des mots clés (tels « Table des matières » ou « Bibliographie »), de la typographie ou de la coupure des mots. La solution standard provient du paquetage **babel** (Braams et Bezos, 2023). Celui-ci permet de combiner plusieurs langues dans un même document et de passer de l'une à l'autre facilement. Consultez la [documentation](#)  du paquetage **babel-french** pour connaître toutes les adaptations au français offertes par ce paquetage incontournable.

📖 frenchb

2.10.4 Guillemets français

Une espace fine devrait séparer les guillemets du ou des mots qu'ils entourent. Pour que l'espace autour des symboles soit géré automatiquement par **babel**, il suffit d'ajouter dans le préambule la commande

```
\frenchbsetup{og=« , fg=»}
```

2.10.5 Séparateur décimal

Le séparateur décimal en français est la virgule et non le point. Or, en mode mathématique, \LaTeX ajoute automatiquement une espace fine après une virgule comme s'il s'agissait d'une énumération.

Afin de pouvoir utiliser de manière conviviale la virgule comme séparateur décimal en mode mathématique, chargez dans vos documents le très pratique paquetage **icomma** (Schmidt, 2002). Avec ce paquetage, la virgule agira comme séparateur décimal en mode mathématique seulement lorsqu'elle est suivie d'un caractère autre que l'espace.

```
% nombre décimal
$x = 1,2$
```

$x = 1,2$

```
% énumération
$x = 1, 2, 3$
```

$x = 1, 2, 3$

TAB. 2.2 - Principaux enjeux de rédaction en français et paquetages offrant des solutions

Enjeu	Solution
Traduction des mots clés prédéfinis	babel
Coupure de mots	babel
Typographie française	babel
Lettres accentuées dans source	source en UTF-8 (X _Y LaTeX) inputenc (LaTeX)
Lettres accentuées dans sortie	fontspec (X _Y LaTeX) fontenc (LaTeX)
Virgule comme séparateur décimal	icomma
Espace comme séparateur des milliers	numprint et babel



Le paquetage **icomma** peut entrer en conflit avec certains autres. Aussi est-il préférable de le charger parmi les derniers dans le préambule, et certainement après les paquetages **fontspec** et **unicode-math** avec X_YLaTeX.

2.10.6 Séparateur des milliers

Le séparateur des milliers en français est l'espace. Lorsque le paquetage **numprint** (Harders, 2012) est chargé, **babel** fournit la commande `\nombre` pour formater automatiquement les nombres.

```
\nombre{123456789} 123 456 789
```

Le [tableau 2.2](#) résume les divers enjeux liés à la rédaction en français avec LaTeX et les paquetages qui offrent des solutions.

2.11 Exercices

2.1 Utiliser le fichier `exercice-minimal.tex`.

- a) Compiler le document avec la classe `article`, puis avec la classe `book`. Observer le résultat.

b) Ajouter du texte en français (avec accents), puis compiler et observer le résultat.

2.2 Modifier le fichier `exercice-commandes.tex` afin de produire le texte ci-dessous.

Les commandes \LaTeX débutent par le symbole `\` et se terminent par le premier caractère autre qu'une lettre, y compris l'espace. Cela a pour conséquence qu'une espace immédiatement après une commande sans argument sera *avalée* par la commande.

La portée d'une commande est **limitée** à la zone entre accolades.

1. L'environnement `enumerate` permet de créer une liste numérotée.
2. Les environnements de listes sont parmi les plus utilisés en \LaTeX .

2.3 a) Compiler le fichier `exercice-classe+paquetages.tex`.

b) Changer la police du document pour 11 points, puis 12 points. Observer l'effet sur les marges et sur la coupure automatique des mots dans le document compilé.

c) Activer le paquetage `icomma` en supprimant le symbole `%` au début de la ligne dans le préambule. Compiler le document et observer l'effet sur la formule mathématique.

d) Charger le paquetage `numprint` avec l'option `auto language` (après le paquetage `babel`). Dans le code source de la formule mathématique, changer

```
| 10 000
```

pour

```
| \nombre{10000}
```

Compiler et observer le résultat.

3 Organisation d'un document

La maîtrise des notions du chapitre précédent permet déjà de composer un document simple avec \LaTeX . Toutefois, la puissance du système de mise en page se manifeste vraiment lors de la préparation de documents élaborés comportant plusieurs divisions internes, une table des matières, des renvois, etc. Le présent chapitre aborde ces aspects d'organisation d'un document.

3.1 Choix d'une classe

La première chose à faire au moment de se lancer dans la rédaction d'un document avec \LaTeX consiste normalement à choisir une classe. Vous avez déjà appris à la [section 2.4](#) comment spécifier la classe à utiliser. Cette section présente les différences entre les classes ainsi que les principales options disponibles.

Les classes standards sont `article`, `report`, `book`, `letter` et `slides`.

`article` Articles scientifiques et autres documents de longueur modérée ne nécessitant pas une mise en page élaborée. Le folio (numéro de page) est placé au centre du pied de page. Le titre apparaît dans le haut de la première page, immédiatement suivi du texte.

`report` Rapports et autres documents plus longs pouvant être divisés en chapitres. Le titre apparaît sur une page de titre. La mise en page est autrement identique à celle de la classe `article`.

`book` Longs documents divisés en chapitres. La mise en page est conçue pour une impression recto verso. L'entête de la page (autre que la première du chapitre) contient le folio sur le bord extérieur et le titre de chapitre (page paire) ou le titre de section (page impaire). Le titre apparaît sur une page de titre.

`letter` Lettres et correspondance. Bien que puissante, cette classe est plus rarement utilisée. Je n'en traite pas davantage dans ce document.

TAB. 3.1 – Caractéristiques des principales classes standards

Classe	Divisions	Disposition	Entête	Pied de page
article	parties, sections, ...	recto	vide	folio centré
report	parties, chapitres, sections, ...	recto	vide	folio centré
book	parties, chapitres, sections, ...	recto verso	folio, titres	vide

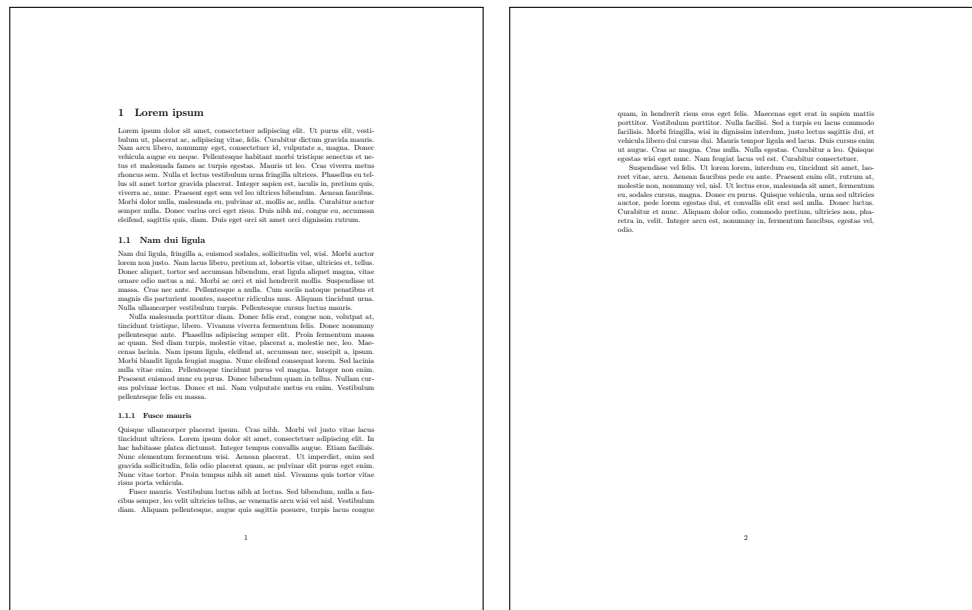


FIG. 3.1 – Exemple de mise en page avec la classe article

slides Diapositives simples pour des présentations. La [section 10.7](#) traite plus en détail de la production de diapositives.

Le [tableau 3.1](#) fournit un sommaire des principales caractéristiques des classes article, report et book. De plus, les figures [3.1-3.3](#) fournissent des exemples de mise en page pour ces trois classes.

Cet ouvrage fait une large place à la classe qui est utilisée pour le com-

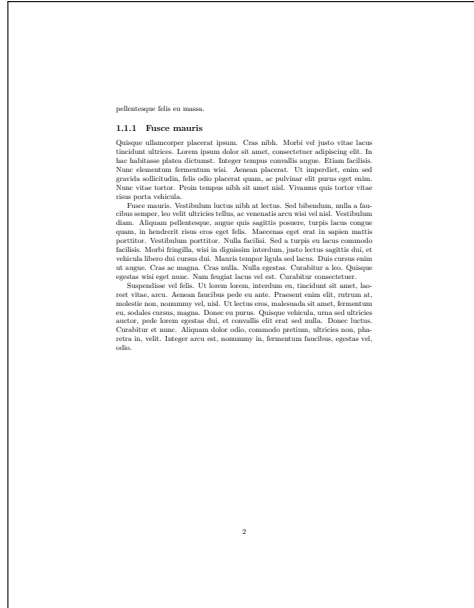
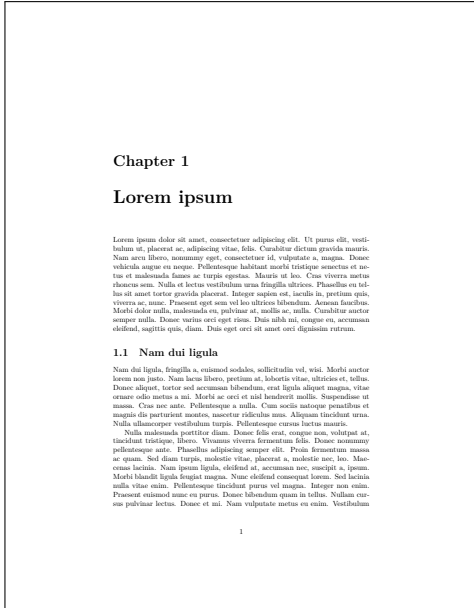


FIG. 3.2 – Exemple de mise en page avec la classe report

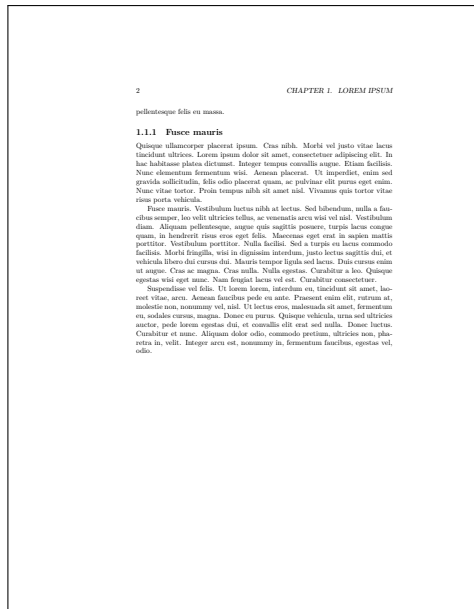
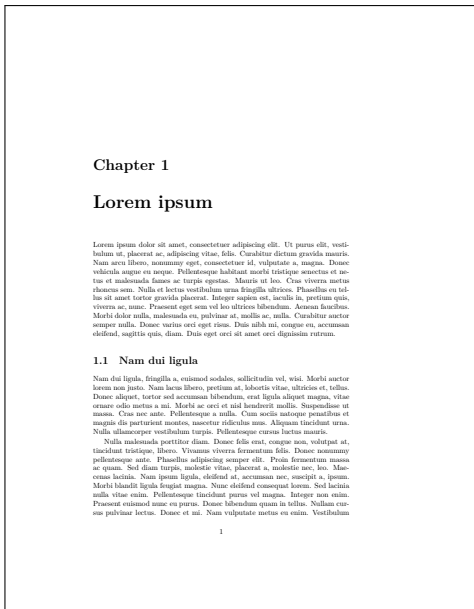


FIG. 3.3 – Exemple de mise en page avec la classe book

poser : la classe `memoir` (Wilson, 2022). Il s'agit d'une extension de la classe standard `book` qui facilite à plusieurs égards la préparation de documents d'allure professionnelle dans \LaTeX . Je recommande d'utiliser cette classe en lieu et place de la classe `book`, ou même de la classe `article` (voir ci-dessous).

La classe `memoir` incorpore d'office plus de 30 des paquetages les plus populaires¹. La classe fait partie des distributions \LaTeX modernes ; elle devrait être installée et disponible sur tout système. Elle est livrée avec une **documentation** exhaustive : le manuel d'instructions fait près de 600 pages ! Il peut être utile de s'y référer de temps à autre pour réaliser une mise en page particulière.

📖 `memoir`

Rappelons que l'on charge une classe de document au début du préambule avec la commande

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

Les *options* disponibles varient d'une classe à l'autre. Les plus courantes sont les suivantes.

`10pt`, `11pt`, `12pt` Taille de la police du document en points. La valeur par défaut est `10pt`. Je recommande d'utiliser plutôt `11pt`.

`oneside`, `twoside` Disposition du document en recto seulement ou en recto verso. Ces options ne sont utiles que pour modifier la disposition par défaut de la classe.

`openright`, `openany` Position de la première page des chapitres toujours à droite (page impaire) ou immédiatement après la dernière page du chapitre précédent. Avec la valeur par défaut, `openany`, \LaTeX insérera une page blanche dans le document si un chapitre se termine sur une page impaire.

`article` (classe `memoir` seulement) Mise en page comme celle d'un article. Avec cette option, `memoir` peut remplacer la classe `article`, ce qui permet d'utiliser une seule et même classe pour les deux principaux types de document (article et livre).

D'autres options permettent de contrôler la position du titre, la disposition en une ou deux colonnes, ou encore la position des équations hors paragraphe. Thurnherr (2014) offre une présentation succincte des options standards. Le chapitre 1 de la documentation de `memoir` traite en plus des ajouts propres à cette classe.

1. Consultez la section 18.24 de la documentation de `memoir` pour la liste ou encore le journal de la compilation (*log*) d'un document utilisant la classe.

3.2 Parties d'un document

Tout document de plus de quelques pages est normalement divisé en chapitres, sections, sous-sections, etc. Il peut comporter une ou plusieurs annexes et débiter par un résumé, notamment s'il s'agit d'un article scientifique. Le document est habituellement coiffé d'un titre, mais celui-ci est parfois affiché sur une page de titre séparée.

Toutes ces considérations relevant essentiellement de la mise en page, \LaTeX s'en charge pour vous. Vous n'avez qu'à spécifier la structure logique du document à l'aide des commandes de la présente section.

3.2.1 Titre et page de titre

\LaTeX rend très simple la composition du titre d'un article scientifique ou d'une page de titre simple (classes report, book, memoir).

En premier lieu, vous devez spécifier, habituellement dans le préambule, le titre du document, le nom du ou des auteurs et la date de publication avec les commandes suivantes :

```
\title{\i{Titre du document}}
\author{\i{Prénom Nom \ \ Affiliation \ \ Adresse}}
\date{\i{Date ou autre texte}}
```

Un long titre sera scindé automatiquement. Vous pouvez aussi scinder le titre manuellement en insérant la commande « \backslash » aux points de coupure. Si l'ouvrage comporte deux auteurs ou plus, insérez les informations dans la commande \backslash author les unes après les autres en séparant chaque entrée par la commande \backslash and. La commande \backslash date insère le texte donné en argument (qu'il s'agisse d'une date ou non) à l'endroit prévu à cet effet par \LaTeX . Si l'on omet la commande, \LaTeX insère la date du jour au moment de la compilation. Pour ne pas afficher la date, laissez simplement l'argument vide :

```
\date{}
```

Dans les articles scientifiques, le nom d'un auteur est fréquemment suivi d'un appel de note renvoyant à des remerciements à un organisme subventionnaire ou à quelque autre information sur l'auteur. On insère une telle note et son appel à l'endroit approprié dans les commandes \backslash title ou \backslash author avec la commande \backslash thanks :

```
\thanks{\i{Texte}}
```

Les commandes ci-dessus ne permettent que de saisir les informations relatives au titre. Pour produire le titre il faut, en second lieu, insérer dans le

corps du document la commande `\maketitle`. La commande insère le titre à l'endroit où elle apparaît dans le code source.

Exemple 3.1. Le code ci-dessous permet de créer un titre d'article standard. La page composée avec \LaTeX (puisque nous utilisons le paquetage **fontspec**) se trouve à la [figure 3.4](#).

```
\documentclass[11pt, french]{article}
\usepackage{fontspec}
\usepackage{babel}
\usepackage{lipsum}

\title{Lorem ipsum dolor sit amet}
\author{Vincent Goulet%
\thanks{Mauris pharetra leo lacus, et suscipit
nulla consequat eget.} \\
École d'actuariat \\
Université Laval}

\begin{document}


\maketitle

\lipsum[1-2]

\end{document}
```

La date de publication qui apparaît dans la page composée est celle de la compilation puisque la commande `\date` n'apparaît pas dans le code source.

Remarquez également que j'ai placé un symbole de commentaire « % » immédiatement après le nom de l'auteur dans le code source. Tel qu'expliqué à la [section 2.9.1](#), c'est pour éviter que \LaTeX ne transforme le retour à la ligne avant la commande `\thanks` en une espace entre le nom et l'appel de note.

(Le paquetage **lorem** utilisé dans cet exemple permet d'insérer du faux texte **Lorem Ipsum**  dans un document \LaTeX .) □

J'ai mentionné plus haut que \LaTeX peut aussi produire automatiquement la page de titre d'un rapport ou d'un livre. Il est toutefois peu probable qu'elle convienne, surtout dans le cas d'un livre. Des options plus flexibles existent : les environnements `titlepage` (classes standards) et `titlingpage` (classe



FIG. 3.4 - Illustration d'un titre d'article standard.

memoir). Ces environnements permettent de définir librement une page de titre :

<pre>\begin{titlepage} <Texte de la page de titre> \end{titlepage}</pre>	<pre>\begin{titlingpage} <Texte de la page de titre> \end{titlingpage}</pre>
--	--

Vous contrôlez alors entièrement la disposition et la composition des éléments de la page de titre. Consultez le chapitre 4 de la [documentation](#) de memoir pour une liste de bonnes pratiques en matière de composition de page de titre et pour des exemples détaillés.

3.2.2 Résumé

Les articles scientifiques et les rapports comportent souvent un résumé, habituellement composé en retrait des marges gauche et droite et dans une police plus petite. Le résumé est produit avec l'environnement `abstract` des classes `article`, `report` ou `memoir` :

```
\begin{abstract}
  <Texte du résumé>
\end{abstract}
```

3.2.3 Sections

Vous pouvez diviser un document en sections qui seront automatiquement numérotées par \LaTeX de manière séquentielle avec les commandes suivantes :

```
\part[<titre court>]{<titre>}
\chapter[<titre court>]{<titre>}
\section[<titre court>]{<titre>}
\subsection[<titre court>]{<titre>}
\subsubsection[<titre court>]{<titre>}
\paragraph[<titre court>]{<titre>}
\subparagraph[<titre court>]{<titre>}
```

Les commandes forment, dans l'ordre ci-dessus, une hiérarchie des titres d'un document². Tel que mentionné précédemment, la commande `\chapter` n'est pas disponible avec la classe `article`.

Chaque commande prend en argument obligatoire le *<titre>* de la section. Si celui-ci est très long, il peut être utile de fournir en argument optionnel un *<titre court>* ; c'est ce dernier qui apparaîtra dans la table des matières et dans les entêtes de page, le cas échéant.

Toutes les commandes existent en version étoilée (nom suivi de « * ») qui supprime la numérotation ainsi que l'insertion éventuelle dans la table des matières (plus de détails à la [section 3.3](#)).



Évitez d'utiliser des sous-sous-sections numérotées (commande `\subsubsection`) dans un livre. Cela résulte en une numérotation à quatre niveaux qui s'avère difficile à suivre pour le lecteur.

2. Je n'ai jamais utilisé les niveaux de division `\paragraph` et `\subparagraph`.

TAB. 3.2 - Commandes d'identification de la structure logique d'un livre et leurs effets

Commande	Effets
<code>\frontmatter</code>	numérotation des pages en chiffres romains (i, ii, ...) chapitres non numérotés
<code>\mainmatter</code>	numérotation des pages à partir de 1 en chiffres arabes chapitres numérotés
<code>\backmatter</code>	numérotation des pages se poursuit chapitres non numérotés

3.2.4 Annexes

Les annexes sont des sections ou des chapitres avec une numérotation alphanumérique (A, A.1, ...) plutôt qu'entièrement numérique. On informe \LaTeX que les sections suivantes doivent être traitées comme des annexes en insérant dans le document la commande

```
\appendix
```

En plus de modifier le style de numérotation, la commande a pour effet de changer le mot clé « Chapitre » pour « Annexe » dans les titres de chapitre.

3.2.5 Structure logique d'un livre

Un livre se compose normalement de trois grandes parties logiques : les pages liminaires (avant-propos, table des matières, tout ce qui précède le chapitre premier); le corps du livre (chapitres et annexes); les parties en fin d'ouvrage (bibliographie, index). Les commandes

```
\frontmatter  
\mainmatter  
\backmatter
```

qui sont disponibles avec les classes book et memoir, permettent d'identifier ces trois parties. Le [tableau 3.2](#) résume l'effet de chaque commande.

3.3 Table des matières

Dans la mesure où vous avez bien identifié les différentes divisions d'un ouvrage avec les commandes mentionnées à la section précédente, la pro-

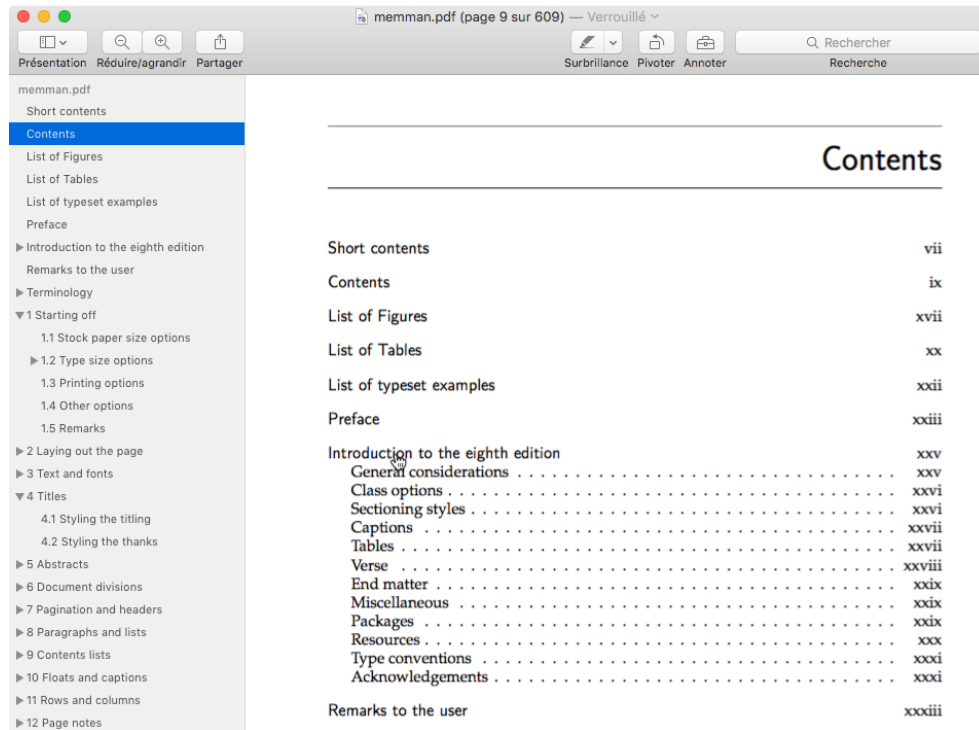


FIG. 3.5 – Consultation d'un document PDF avec la visionneuse Aperçu de macOS. La barre latérale de gauche affiche la table des matières du fichier PDF, ce qui permet de naviguer dans le document sans devoir revenir à celle du document.

duction de la table des matières est on ne peut plus simple avec \LaTeX : il suffit d'insérer la commande

```
\tableofcontents
```

dans le corps du document à l'endroit où la table des matières doit apparaître. C'est tout!

Lorsque le paquetage **hyperref** (Rahtz et Oberdiek, 2023) est chargé, la commande `\tableofcontents` produit également la table des matières du fichier PDF. Cela permet de naviguer dans le document directement depuis la visionneuse PDF. La figure 3.5 illustre cette fonctionnalité avec la visionneuse Aperçu de macOS.



La production initiale de la table des matières et la prise en compte de toute modification requiert jusqu'à trois compilations consécutives du document.

Par défaut, une section non numérotée ne figure pas dans la table des matières. Si l'y insérer néanmoins, vous devez utiliser la commande suivante :

```
\addtocontents{toc}{\langle niveau \rangle \langle titre \rangle}
```

où *⟨niveau⟩* est le nom de la commande de division sans le caractère `\` (`chapter`, `section`, etc.) et *⟨titre⟩* est le texte qui doit figurer dans la table des matières.

Outre une table des matières, les ouvrages scientifiques comportent parfois une liste des figures et une liste des tableaux. \LaTeX les produit automatiquement avec les commandes

```
\listoffigures
\listoftables
```

Dans la classe `memoir`, les commandes ci-dessus insèrent leur propre titre de section dans la table des matières (autrement dit, la table des matières apparaît dans la table des matières). Les versions

```
\tableofcontents*
\listoffigures*
\listoftables*
```

adoptent le comportement des classes standards, soit d'omettre ces parties dans la table des matières.

3.4 Renvois automatiques

Un document d'une certaine ampleur contiendra souvent des renvois à une section, un tableau, une équation, voire une page spécifique. Évidemment, le numéro de la section, du tableau, de l'équation ou de la page est susceptible de changer au fil de la rédaction du document. C'est pourquoi vous ne devriez *jamais* insérer les renvois manuellement dans le texte. C'est une tâche qu'il vaut mieux confier à l'ordinateur.

3.4.1 Étiquettes et renvois

Les renvois automatiques dans \LaTeX reposent sur un système d'étiquettes attribuées à des éléments de contenu et de référencement par ces étiquettes.

Ainsi, pour effectuer un renvoi vers un élément de contenu, il faut d'abord nommer celui-ci en insérant la commande

```
| \label{<nom>}
```

à proximité de l'élément. Le choix du <nom> est tout à fait libre. Il peut être constitué de toute combinaison de lettres, de chiffres et de symboles autres que les symboles réservés mentionnés à la [section 2.9.2](#).

Ensuite, il devient possible d'insérer un renvoi à cet élément de contenu n'importe où dans le document avec la commande

```
| \ref{<nom>}
```

Pour renvoyer à la page où se trouve l'élément, utilisez la commande

```
| \pageref{<nom>}
```

Exemple 3.2. Le code ci-dessous permet d'insérer un renvoi automatique à un numéro de section dans un document. Le résultat se trouve à la [figure 3.6](#).

```
\section{Définitions}
\label{sec:definitions}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Duis in auctor dui. Vestibulum ut, placerat ac, adipiscing
vitae, felis.

\section{Historique}

Tel que vu à la section \ref{sec:definitions}, on a...
```

□

Un long document contiendra vraisemblablement un grand nombre d'étiquettes et de renvois. Afin de vous y retrouver et pour réduire les risques de doublons, adoptez une manière systématique et mnémotechnique de nommer les éléments. Par exemple, pour le présent document, j'ai utilisé un système d'étiquettes de la forme suivante :

- ▶ chap:<chapitre> pour les chapitres;
- ▶ sec:<chapitre>:<section> pour les sections;
- ▶ tab:<chapitre>:<tableau> pour les tableaux;
- ▶ eq:<chapitre>:<equation> pour les équations.

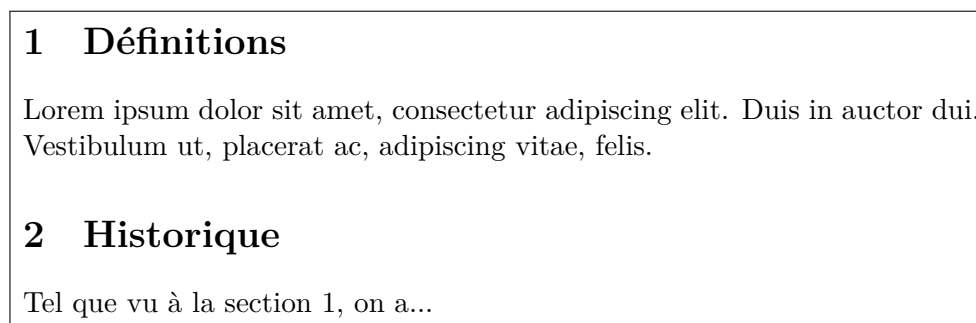


FIG. 3.6 - Texte produit par le code de l'exemple 3.2 illustrant un renvoi automatique standard

3.4.2 Production des renvois

La production des renvois requiert deux à trois compilations. Tant que \LaTeX n'a pas complété les renvois, le journal de compilation contient, vers la toute fin, le message

```
LaTeX Warning: Label(s) may have changed. Rerun to get  
cross-references right.
```

Prenez également garde aux alertes suivantes dans le journal de compilation. Elles identifient des correctifs à apporter dans le système d'étiquettes et de renvois. D'abord, le message

```
LaTeX Warning: There were undefined references.
```

indique qu'une ou plusieurs commandes `\ref` renvoient à un $\langle nom \rangle$ qui n'est pas défini avec `\label`. Le document contiendra alors le caractère ? en lieu et place du renvoi. À l'inverse, le message

```
LaTeX Warning: There were multiply-defined labels.
```

indique que plusieurs commandes `\label` utilisent le même $\langle nom \rangle$. Cela est susceptible de causer des renvois vers le mauvais élément de contenu.

3.4.3 Renvois avec hyperliens

Lorsque chargé dans le document, le paquetage **hyperref** insère des hyperliens vers les renvois dans les fichiers PDF. C'est très pratique en consultation électronique d'un document.

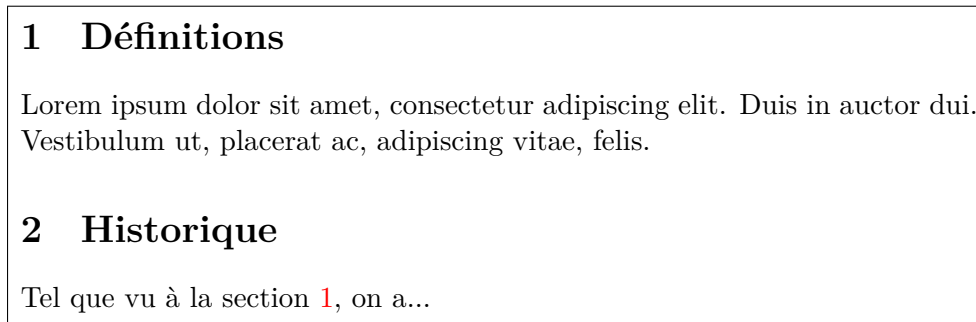


FIG. 3.7 - Texte produit par le code de l'exemple 3.2 après l'ajout du paquetage **hyperref**

Exemple 3.3. Avec l'ajout de

```
\usepackage{hyperref}
```

dans le préambule, le code de l'exemple 3.2 produit maintenant le type de renvoi illustré à la figure 3.7. Le texte en couleur contrastante est un hyperlien vers le titre de la section. □

L'inconvénient avec la procédure illustrée dans l'exemple précédent, c'est que seul le numéro de la référence est transformé en hyperlien. La zone disponible pour cliquer s'en trouve plutôt restreinte. Une fonctionnalité du paquetage **hyperref** — à laquelle j'ai eu recours dans le présent document — permet d'agrandir cette zone. La commande

```
\autoref{<nom>}
```

permet de nommer automatiquement le type de renvoi (section, équation, tableau, etc.) et de transformer en hyperlien à la fois ce texte et le numéro du renvoi.

Exemple 3.4. Reprenons le texte de l'exemple 3.2, mais en utilisant cette fois la commande `\autoref` pour insérer un renvoi dans le texte. Remarquez que le mot « section » a été supprimé du code source pour laisser à la commande le soin d'insérer l'étiquette appropriée dans le document.

```
\section{Définitions}
\label{sec:definitions}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Duis in auctor dui. Vestibulum ut, placerat ac, adipiscing
```

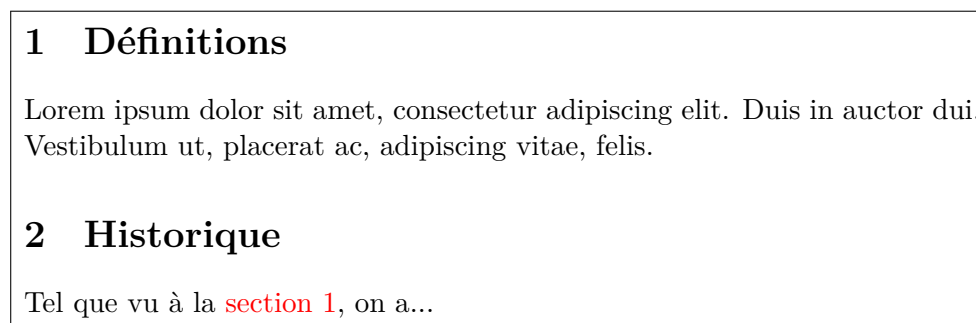


FIG. 3.8 - Texte produit par le code de l'exemple 3.4

```
vitae, felis.  
  
\section{Historique}  
  
Tel que vu à la \autoref{sec:definitions}, on a...
```

La [figure 3.8](#) montre que le mot « section » ainsi que son numéro forment maintenant l'hyperlien. □

La [section 10.4](#) fournit des détails additionnels sur la gestion des hyperliens dans un document PDF.

3.5 Document contenu dans plusieurs fichiers

Lorsque le préambule et le corps du texte demeurent relativement courts (peu de commandes spéciales et moins d'une vingtaine de pages de texte), il demeure assez simple et convivial d'en faire l'édition dans un seul fichier à l'aide de votre éditeur de texte favori.

Cependant, si le préambule devient long et complexe ou, surtout, lorsque l'ampleur du document augmente jusqu'à compter un grand nombre de pages sur plusieurs chapitres, il convient de répartir les diverses parties du document dans des fichiers séparés.

La segmentation en plusieurs fichiers rend l'édition du texte plus simple et plus efficace. De plus, durant la phase de rédaction, elle peut significativement accélérer la compilation des documents très longs ou comptant plusieurs images.

3.5.1 Insertion du contenu d'un autre fichier

La commande `\input` permet d'insérer le contenu d'un autre fichier dans un document \LaTeX . La syntaxe de la commande est

```
\input{\fichier}
```

où le nom du fichier à insérer est `\fichier.tex`. L'extension `.tex` est donc implicite. Le contenu du fichier est inséré tel quel dans le document, comme s'il avait été saisi dans le fichier qui contient l'appel à `\input`.

Le procédé est surtout utile pour sauvegarder séparément des bouts de code qui gênent l'édition du texte (figures, longs tableaux) ou qui sont communs entre plusieurs documents (licence d'utilisation, auteurs et affiliations).

La commande peut aussi être utilisée dans le préambule pour charger une partie ou l'ensemble de celui-ci. Cela permet de composer un même préambule pour plusieurs documents. Il suffit alors de faire d'éventuelles modifications à un seul endroit pour les voir prendre effet dans tous les documents.

3.5.2 Insertion de parties d'un document

Je vous recommande de segmenter tout document d'une certaine ampleur dans des fichiers `.tex` distincts pour chaque partie — habituellement un fichier par chapitre. Vous composerez ensuite le document complet à l'aide d'un fichier maître qui contient le préambule \LaTeX et un ensemble d'appels à la commande `\include` pour réunir les parties dans un tout.

La syntaxe de `\include` est

```
\include{\fichier}
```

où le nom du fichier à insérer est `\fichier.tex`. Ici aussi l'extension `.tex` est implicite.

Comme `\input`, la commande `\include` insère le contenu d'un autre fichier dans un document. Toutefois, l'insertion d'un fichier avec `\include` débute toujours une nouvelle page. Cette commande sert donc principalement pour insérer des chapitres entiers plutôt que seulement des portions de texte. De plus, un fichier inséré avec `\include` peut contenir des appels à `\input`, mais pas à `\include`.

Exemple 3.5. La [figure 3.9](#) présente un exemple type de fichier maître qui fait appel à la commande `\include` pour composer un document chapitre par chapitre. □


```
1 \documentclass{memoir}
2   [...]
3
4 \begin{document}
5
6 \frontmatter
7
8 \include{introduction}
9 \tableofcontents*
10
11 \mainmatter
12 \include{historique}
13 \include{rappels}
14 \include{modele}
15   [...]
16
17 \end{document}
```

FIG. 3.9 – Structure type d'un fichier maitre. Les fichiers `historique.tex`, `rappels.tex` et `modele.tex` contiennent le texte des trois premiers chapitres.

3.5.3 Compilation partielle

Le principal avantage de `\include` par rapport à `\input` réside dans le fait que \LaTeX peut préserver entre les compilations les informations telles que les numéros de pages, de sections ou d'équations, ainsi que les renvois et les références bibliographiques. Cela permet, par exemple, de compiler le texte d'un seul chapitre — plutôt que le document entier — et de néanmoins obtenir une image représentative du chapitre. Procéder ainsi accélère significativement la compilation des documents longs ou complexes.

La commande `\includeonly`, qui est employée exclusivement dans le préambule, sert à spécifier le ou les fichiers à compiler tout en préservant la numérotation et les références. Sa syntaxe est

```
\includeonly{<liste_fichiers>}
```

où `<liste_fichiers>` contient les noms des fichiers à inclure dans la compilation, séparés par des virgules et sans l'extension `.tex`.

Lors de l'utilisation de la commande `\includeonly`, toute la numérotation

tion dans les fichiers $\langle liste_fichiers \rangle$ suivra celle établie lors de la compilation précédente. Si l'édition des fichiers de $\langle liste_fichiers \rangle$ cause des changements dans la numérotation et les références dans les autres parties du document, une nouvelle compilation de l'ensemble ou d'une partie de celui-ci s'avérera nécessaire.

Exemple 3.6. Un document est composé en plusieurs parties avec les commandes suivantes :

```
\include{historique}           % chapitre 1
\include{rappels}              % chapitre 2
\include{modele}               % chapitre 3
```

Les chapitres débutent respectivement aux pages 1, 23 et 41.

- Si l'on ajoute au préambule du document la commande

```
\includeonly{rappels}
```

le numéro du chapitre sera toujours 2 et le folio de la première page sera toujours 23, même si les 22 pages précédentes ne se trouvent pas dans le document.

- Si l'on modifie le fichier `rappels.tex` de telle sorte que le chapitre se termine maintenant à la page 46, il faudra recompiler le document avec au moins les fichiers `rappels.tex` et `modele.tex` pour que les pages du chapitre 3 soient numérotées à partir de 47.

□

L'[exercice 3.3](#) illustre mieux le cycle typique d'utilisation des commandes `\include` et `\includeonly`.



Utilisez des noms de fichiers qui permettent de facilement identifier leur contenu. Par exemple, un nom comme `rappels.tex` identifie clairement le contenu du fichier et il résiste mieux aux changements à l'ordre des chapitres que `chapitre1.tex`.

3.6 Exercices

3.1 Utiliser le fichier `exercice-parties.tex`.

- Étudier la structure du document dans le code source.
- Ajouter un titre et un auteur au document à l'aide des commandes `\title` et `\author` se trouvant déjà dans le préambule.

- c) Créer la table des matières du document en le compilant deux à trois fois.
- d) Insérer deux ou trois titres de sections de différents niveaux dans le document et recompiler.
- e) La numérotation cesse à partir des sous-sections. C'est une particularité de la classe `memoir`. Recompiler le document après avoir ajouté au préambule la commande

```
| \maxsecnumdepth{subsection}
```

- f) Ajouter une annexe au document.

3.2 Utiliser le fichier `exercice-renvois.tex`.

- a) Insérer dans le texte un renvoi au numéro d'une section.
- b) Activer le paquetage `hyperref` avec l'option `colorlinks` et comparer l'effet d'utiliser `\ref` ou `\autoref` pour le renvoi.

3.3 Cet exercice fait appel au fichier maître `exercice-include.tex` et à plusieurs fichiers auxiliaires. Schématiquement, le document est composé ainsi :

```
exercice-include.tex
├─ \input pagetitre.tex
├─ \include rpresentation.tex
│   └─ \includegraphics console-screenshot.pdf
└─ \include emacs.tex
```

La commande `\includegraphics` permet d'insérer une image dans un document \LaTeX . Elle provient du paquetage `graphicx`.

- a) Étudier le code source du fichier maître, puis le compiler deux à trois fois jusqu'à ce que tous les renvois soient à jour. Il est normal à ce stade que la figure 1 du document soit vide.
- b) Ajouter dans le préambule du fichier maître la commande

```
| \includeonly{emacs}
```

puis compiler le document.

Observer que, malgré l'absence du chapitre 1, la numérotation et les références demeurent à jour, notamment la table des matières.

- c) Remplacer la commande ajoutée en b) dans le préambule du fichier maître par la commande

```
| \includeonly{rpresentation}
```

Vers la fin du fichier `rpresentation.tex`, activer la commande

```
\includegraphics[width=\textwidth]{console-screenshot}
```

en supprimant le symbole « % » au début de la ligne. Compiler de nouveau le document deux fois.

Les modifications ont eu pour effet d'ajouter une page au chapitre 1. Observer que, selon la table des matières, le chapitre 2 débute toujours à la page 3 alors que celle-ci est maintenant occupée par la figure 1.

- d) Afin de corriger la table des matières, désactiver dans le préambule du fichier `maitre` la commande `\includeonly`, puis compiler de nouveau le document quelques fois.

- 3.4 Déplacer dans un fichier `preambule.tex` toutes les lignes du préambule du fichier `exercice-include.tex` utilisé à l'exercice précédent, à l'exception de celles relatives à la page titre (titre, auteur, date). Insérer le préambule dans `exercice-include.tex` avec la commande `\input`.

4 Apparence et disposition du texte

Les bonnes pratiques dictent de séparer le contenu du texte de son apparence lorsque l'on utilise un système de mise en page comme \LaTeX . Néanmoins, vous pourriez souhaiter modifier l'apparence générale du document ou disposer le texte d'une manière particulière. \LaTeX offre toute la flexibilité voulue, que ce soit pour contrôler les attributs et la taille des polices de caractères, disposer du texte sous forme de liste numérotée ou à puce, centrer du texte ou créer des notes de base de page et des citations.

4.1 Police de caractères

Par défaut, tous les documents \LaTeX utilisent la même police de caractères¹, Computer Modern. Chose qui ne manque jamais de surprendre les utilisateurs débutants : le système n'a pas été conçu pour changer facilement la police du document !

Cela dit, il est aujourd'hui devenu assez simple d'utiliser d'autres polices pour vos documents, surtout avec les moteurs \TeX modernes comme \XeTeX . La [section 10.2](#) traite du sujet plus en détail.

Cette section se concentre plutôt sur le changement d'*attribut* de la police du document, qu'il s'agisse de la famille (avec ou sans empattements, à largeur fixe ou variable), de la forme (droit, italique, penché) ou de la graisse (normal, gras). Le [tableau 4.1](#) dresse la liste des commandes de changement d'attribut de la police.

La commande `\normalfont` permet de revenir d'un trait à la police par défaut, soit le romain, droit de graisse moyenne.

Exemple 4.1. La commande `\setsecheadstyle` de la classe `memoir` permet de modifier facilement le style des titres de section pour tout le document. La commande suivante placée dans le préambule permet d'obtenir des titres de section en gras sans empattements :

1. Donald Knuth a créé la police en même temps que \TeX .

TAB. 4.1 - Commandes de changement d'attribut de la police. Les commandes de la deuxième colonne s'appliquent à tout le texte qui suit. Celles de la troisième colonne s'appliquent uniquement au texte en argument.

Famille		
romain	<code>\rmfamily</code>	<code>\textrm{<texte>}</code>
largeur fixe	<code>\ttfamily</code>	<code>\texttt{<texte>}</code>
sans empattements	<code>\sffamily</code>	<code>\textsf{<texte>}</code>
Forme		
droit	<code>\upshape</code>	<code>\textup{<texte>}</code>
<i>italique</i>	<code>\itshape</code>	<code>\textit{<texte>}</code>
<i>penché</i> [†]	<code>\slshape</code>	<code>\textsl{<texte>}</code>
PETITES CAPITALES	<code>\scshape</code>	<code>\textsc{<texte>}</code>
Série		
moyen	<code>\mdseries</code>	<code>\textmd{<texte>}</code>
gras	<code>\bfseries</code>	<code>\textbf{<texte>}</code>

[†] diffère de l'italique selon la police utilisée

```
\setseheadstyle{\normalfont\sffamily\bfseries}
```

La commande `\normalfont` au début de l'argument permet de réinitialiser le style des titres, question d'éviter d'éventuels conflits avec une configuration antérieure. □

4.2 Taille du texte

Vous vous souviendrez que l'on règle la taille de base du texte au chargement de la classe du document, tel qu'expliqué à la [section 2.4](#). Les commandes du [tableau 4.2](#) permettent néanmoins de réduire ou d'agrandir la taille des caractères pour une section du texte. Les commandes servent également pour la configuration de l'apparence générale du document dans le préambule.

Exemple 4.2. Les titres de sections sont généralement composés dans une taille supérieure à celle du texte. La commande suivante permet d'augmenter de deux échelons la taille des titres de l'[exemple 4.1](#) :

```
\setseheadstyle{\normalfont\Large\sffamily\bfseries}
```

□

TAB. 4.2 – Commandes de changement de la taille des caractères. Toutes les tailles sont relatives à celle de police de base du document.

<code>\miniscule[†]</code>	minuscule
<code>\tiny</code>	vraiment petit
<code>\scriptsize</code>	encore plus petit
<code>\footnotesize</code>	plus petit
<code>\small</code>	petit
<code>\normalsize</code>	taille normale
<code>\large</code>	grand
<code>\Large</code>	plus grand
<code>\LARGE</code>	encore plus grand
<code>\huge</code>	énorme
<code>\Huge</code>	encore plus énorme
<code>\HUGE[†]</code>	vraiment énorme

[†] ajout de la classe memoir

4.3 Italique

L'italique est l'un des attributs de police les plus fréquemment employés dans le texte. Il sert, notamment, à insister sur des mots, à composer les expressions et locutions en langue étrangère ou à détacher les titres d'œuvres du fil du texte.

Évidemment, il est possible d'obtenir de l'italique avec la commande `\textit` du [tableau 4.1](#). Cependant, je recommande plutôt d'utiliser une commande spécifiquement dédiée à mettre en évidence une portion de texte :

```
\emph{<texte>}
```

Par défaut, la commande `\emph` (pour *emphasis*, « emphase ») placera `<texte>` en italique dans du texte en romain (droit), ou encore en romain dans du texte déjà en italique.

```
C'était un peu \emph{rough}
par moments.
```

C'était un peu *rough* par moments.

```
Il m'a dit : «\emph{Enough
\emph{poutine} for the
week!}»
```

Il m'a dit : « *Enough poutine for the week!* »



Le soulignement servait pour remplacer l'italique à l'ère des dactylos. C'est aujourd'hui une marque de typographie très rarement utilisée, voire à éviter. C'est d'ailleurs la raison pour laquelle il n'existe pas de commande de soulignement dans les classes \LaTeX standards.

4.4 Listes

\LaTeX offre deux environnements principaux pour la composition de listes ou d'énumérations :

```
\begin{itemize}
\item <texte>
\item <texte>
...
\end{itemize}
```

```
\begin{enumerate}
\item <texte>
\item <texte>
...
\end{enumerate}
```

L'environnement `itemize` crée une liste à puce, alors que l'environnement `enumerate` crée une énumération. Il est possible d'imbriquer les listes les unes dans les autres, et ce, peu importe leur type. \LaTeX se chargera d'adapter les marqueurs ou la numérotation jusqu'à quatre niveaux de profondeur.

Exemple 4.3. La liste mixte suivante résume les étapes de création d'une liste avec \LaTeX . Le texte qui a servi à la composer suit la liste.

1. Décider s'il s'agit d'une liste à puce ou d'une énumération;
 - ▶ pour une liste à puce utiliser environnement `itemize`;
 - chaque niveau d'une liste à puce possède un marqueur différent;
 - ▶ pour une énumération utiliser environnement `enumerate`;
2. Débuter chaque élément de la liste par la commande `\item`.
 - a) utiliser simplement un autre environnement `itemize` ou `enumerate` comme texte d'un élément pour créer des listes imbriquées;
 - b) \LaTeX ajustera automatiquement les marqueurs;
3. S'assurer de fermer tous les environnements dans le bon ordre pour retourner au texte normal.

```
\begin{enumerate}
\item Décider s'il s'agit d'une liste à puce ou [...]
  \begin{itemize}
```



```

\item pour une liste à puce utiliser [...]
  \begin{itemize}
  \item chaque niveau d'une liste à puce [...]
  \end{itemize}
\item pour une énumération utiliser [...]
  \end{itemize}
\item Débuter chaque élément de la liste par la commande
  \verb=\item=.
  \begin{enumerate}
  \item utiliser simplement un autre environnement [...]
  \item {\LaTeX} ajustera automatiquement les marqueurs;
  \end{enumerate}
\item S'assurer de fermer tous les environnements [...]
  \end{enumerate}

```

□

TeX permet de configurer à peu près toutes les facettes de la présentation des listes : marqueurs, folios, alignement, espacement entre les éléments, etc. Cette grande flexibilité implique une certaine complexité et la configuration des listes dépasse le cadre du présent document. Plusieurs paquets facilitent toutefois la configuration des listes ; je recommande à ce titre le paquetage **enumitem** (Bezoz, 2019).



Le mode français de **babel** remplace les marqueurs par défaut de l'environnement **itemize** (« • », « - », « * » et « . ») par le tiret cadratin « — ». La documentation de **babel-french** explique comment restituer aux marqueurs les valeurs standards, ou comment choisir vos propres marqueurs.

Il existe un troisième environnement, plus rarement utilisé, pour composer du texte sous une forme s'apparentant à une liste. L'environnement **description** permet d'associer des termes à une définition ou une description pour chacun.

```

\begin{description}
\item[rouge] première couleur   rouge première couleur
\item[vert] deuxième couleur    vert deuxième couleur
\item[bleu] troisième couleur   bleu troisième couleur
\end{description}

```

4.5 Texte centré

On obtient du texte centré avec l'environnement `center` ou à l'aide de la commande `\centering`. L'environnement sert principalement pour centrer un bloc de texte.

<pre>\begin{center} Centrer un mot ou une expression les met en évidence. \end{center}</pre>	<pre>Centrer un mot ou une expression les met en évidence.</pre>
--	--

Remarquez comment le texte centré n'est pas placé en retrait des marges gauche et droite (illustrées par les lignes verticales ci-dessus). Lorsque le texte fait plus d'une ligne, vous pouvez forcer des retours à la ligne avec la commande « `\` ».

<pre>\begin{center} Centrer un mot \ ou une expression \ les met en évidence. \end{center}</pre>	<pre>Centrer un mot ou une expression les met en évidence.</pre>
--	--

La commande `\centering`, quant à elle, centre tout le texte qui la suit. Elle s'avère surtout utile à l'intérieur d'environnements comme `table` ou `figure` pour centrer un tableau ou une figure sur la page. Vous la rencontrerez dans plusieurs des exemples du [chapitre 6](#).

4.6 Citations

Une citation est une reproduction d'un propos ou d'un écrit antérieur d'un auteur. Il existe deux environnements dans \LaTeX pour composer les citations.

L'environnement `quote` est utilisé pour les citations courtes. Le texte est placé en retrait des marges gauche et droite. Le présent texte a été placé dans cet environnement.

Ce paragraphe et les deux suivants sont plutôt composés dans un environnement `quotation`.

Cet environnement sert pour les citations plus longues se comptant en paragraphes.

La principale différence par rapport à l'environnement `quote`, c'est que `quotation` prend aussi en charge les marques de paragraphes.

4.7 Notes de bas de page

Une note de bas de page — ou tout simplement *note*, puisqu'elle peut apparaître ailleurs qu'au bas de la page — est un court texte qui sert à expliquer un mot ou une phrase du texte principal. L'*appel de note* désigne le signe, le chiffre ou la lettre que l'on place après la partie à expliquer.

La commande

```
\footnote{<texte de la note>}
```

insère un appel de note à l'endroit où la commande apparaît dans le texte et affiche *<texte de la note>* à l'endroit approprié². \LaTeX se charge automatiquement de la numérotation et de la disposition des notes.

Pour éviter que l'appel de note ne soit séparé du texte à expliquer, la commande `\footnote` doit immédiatement suivre celui-ci. La gestion des espaces peut alors devenir délicate, surtout si l'appel de note se trouve en fin de phrase. Je recommande de faire bon usage du symbole « % » pour délimiter le contenu de `\footnote` du texte courant.

```
%% note en fin de phrase
```

```
Je remercie Pierre Lasou\footnote{%
  Spécialiste en ressources documentaires.} %
Il fut d'une grande aide dans la préparation de ...
```

```
%% note au fil de la phrase
```

```
Je souligne que Pierre Lasou\footnote{%
  Spécialiste en ressources documentaires.} %
fut d'une grande aide dans la préparation de ...
```

Les usages en matière de numérotation et de disposition des notes varient beaucoup d'une discipline à l'autre : numérotation consécutive pour tout le document, notes dans la marge plutôt qu'au bas de la page, notes rassemblées à la fin de chaque chapitre ou à la toute fin du document, etc. Différents paquetages permettent de réaliser ce type de disposition, notamment

². Au bas de la page par défaut, comme ici. La note est composée dans un corps plus petit et elle est séparée du texte par un trait.

memoir

`footmisc` (Fairbairns, 2022). La classe `memoir` compte également de nombreuses fonctionnalités pour la gestion des notes; consultez le chapitre 12 de la [documentation](#).

4.8 Texte brut et code source

Il est parfois utile d'afficher du texte exactement comme il a été saisi dans un fichier, sans que \LaTeX n'interprète les commandes qui s'y trouvent ou n'en modifie la disposition. C'est ainsi que j'ai composé les nombreux extraits de code du présent document.

L'environnement `verbatim` permet de présenter du texte tel qu'il est entré dans le code source du document.

<pre>\begin{verbatim} Texte disposé exactement tel qu'il est tapé et présenté dans une police à largeur fixe \end{verbatim}</pre>	<pre>Texte disposé exactement tel qu'il est tapé et présenté dans une police à largeur fixe</pre>
---	---

La variante `verbatim*` affiche les espaces de manière explicite en les remplaçant par le symbole `_`. C'est surtout utile pour le code source de langages informatiques où le nombre d'espaces joue un rôle important.

Pour afficher du texte brut ou une commande \LaTeX au fil du texte, on aura recours à la commande `\verb`. Sa syntaxe est un peu particulière :

```
\verb<c><source><c>
```

où `<c>` est un symbole quelconque ne se trouvant pas dans `<source>` et qui sert à délimiter le début et la fin de `<source>`. La variante `\verb*` affiche elle aussi les espaces.

<pre>{\LaTeX} n'interprète pas la commande \verb=\bfseries= lorsqu'elle est placée dans \verb=\verb=.</pre>	<p>\LaTeX n'interprète pas la commande <code>\bfseries</code> lorsqu'elle est placée dans <code>\verb</code>.</p>
---	--

<pre>Le nombre d'espaces entre les mots ne compte pas dans {\LaTeX}: \verb*=a b= et \verb*=a b= donnent «a b».</pre>	<p>Le nombre d'espaces entre les mots ne compte pas dans \LaTeX : <code>a_b</code> et <code>a_ _ b</code> donnent « a b ».</p>
--	--

Pour tout document comportant une grande quantité de texte brut ou de code source, je recommande d'employer le paquetage **listings** (Heinz et collab., 2023). La [section 10.5](#) approfondit le sujet.

4.9 Exercice

- 4.1 a) Ouvrir le fichier `exercice-complet.tex` et en étudier le code source, puis le compiler.
- b) Supprimer l'option `article` au chargement de la classe et compiler de nouveau le document. Observer l'effet de cette option de la classe `memoir`.
- c) Au dernier paragraphe de la première section, placer toute la phrase débutant par «`De simple dérivé`» à l'intérieur d'une commande `\emph` et compiler de nouveau le document.
- d) Changer la puce des listes pour le symbole `$>$` et compiler de nouveau le document.

5 Boîtes

Il arrive que l'on doive traiter de manière spéciale une aire rectangulaire de texte ; pour l'encadrer, la mettre en surbrillance ou la mettre en exergue, par exemple.

Avec les traitements de texte, les tableaux sont souvent mis à contribution à de telles fins. Or, les tableaux devraient être réservés à la disposition d'information sous forme de lignes et de colonnes. Pour disposer et mettre en forme tout autre type contenu se présentant sous forme rectangulaire, \LaTeX offre la solution plus générale des « boîtes ».

Il existe trois sortes de boîtes en \LaTeX : les boîtes horizontales, dont le contenu est disposé exclusivement côte à côte ; les boîtes verticales, qui peuvent contenir plusieurs lignes de contenu ; les boîtes de réglure pour former des lignes pleines de largeur et de hauteur quelconques.

Il n'est pas inutile de savoir, au passage, que \TeX ne manipule que cela, des boîtes. Pour \TeX , chaque caractère, chaque lettre n'est qu'un rectangle d'une certaine largeur qui s'élève au-dessus de la ligne de base (les lignes d'une feuille lignée) et qui, parfois, se prolonge sous la ligne de base (pensons aux lettres « p », « y » ou « Q »). La [figure 5.1](#) illustre cela.

Les commandes et les environnements présentés dans ce chapitre permettent de créer des boîtes dont le contrôle des dimensions et du contenu est laissé entre vos mains.

Une fois créée, une boîte ne peut être scindée en parties, notamment entre les lignes ou entre les pages.

5.1 Boîtes horizontales

Le concept de boîte le plus simple dans \LaTeX est celui de boîte horizontale, c'est-à-dire dont le contenu est disposé latéralement de gauche à droite ¹. Le

1. D'où l'appellation *LR (left-right) box* en anglais.

fantastique

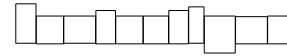


FIG. 5.1 – Pour TeX, un texte n’est qu’un alignement de boîtes. À gauche, ce que nous voyons dans le document fini ; à droite, ce que TeX manipule.

contenu est normalement du texte, mais conceptuellement ce pourrait être n’importe quoi, y compris d’autres boîtes.

Les commandes de base pour créer des boîtes horizontales sont :

```
\mbox{<texte>}
\fbox{<texte>}
```

Elles produisent une boîte de la largeur précise de *<texte>*. Avec la commande `\fbox`, le texte est au surplus encadré.

Il existe également des versions plus générales des commandes `\mbox` et `\fbox` :

```
\makebox[<largeur>][<pos>]{<texte>}
\framebox[<largeur>][<pos>]{<texte>}
```

Les arguments optionnels *<largeur>* et *<pos>* déterminent respectivement la largeur de la boîte et la position du texte dans la boîte. Les valeurs possibles de *<pos>* sont : `l` pour du texte aligné à gauche, `r` pour du texte aligné à droite et `c` (la valeur par défaut) pour du texte centré. Ainsi, la commande

```
\framebox[3.5cm][l]{aligné à gauche}
```

produit aligné à gauche, alors que

```
\makebox[3.5cm]{centré}
```

produit centré.



En usage courant, la commande `\mbox` sert principalement à deux choses :

1. réunir en un bloc du texte que l’on ne veut pas voir scindé entre les lignes ou entre les pages ;
2. créer une boîte vide avec `\mbox{}` afin de laisser croire à TeX que du contenu apparaît à un endroit, sans toutefois qu’il n’occupe aucun espace.

La seconde utilisation fait l’objet de l’[exercice 5.2](#).


Il est parfois nécessaire d'ajuster le positionnement vertical d'éléments de contenu, notamment pour les symboles ou les images. La commande

```
\raisebox{⟨déplacement⟩}{⟨texte⟩}
```

produit une boîte horizontale dont le contenu *⟨texte⟩* est surélevé de la longueur *⟨déplacement⟩* par rapport à la ligne de base. Si *⟨déplacement⟩* est négatif, la boîte est positionnée sous la ligne de base.

```
Texte \raisebox{1ex}{au-dessus} Texte au-dessus de la ligne de  
de la ligne de base. base.
```

```
Texte \raisebox{-1ex}{au-dessous} Texte au-dessous de la ligne de  
de la ligne de base. base.
```

Attention, toutefois, de ne pas utiliser `\raisebox` pour placer du texte en exposant ou en indice. Selon la nature du texte, employez plutôt les commandes `\textsuperscript` et `\textsubscript`, les commandes de la famille `\ieme` de **babel** (section 1.1 de la [documentation](#) ) ou, pour des symboles mathématiques, les commandes d'exposant et d'indice spécifiques au mode mathématique (section 7.3.1).

▀ babel-french

5.2 Boîtes verticales

Les boîtes verticales se distinguent des boîtes horizontales par le fait qu'elles peuvent contenir plusieurs lignes de contenu empilées les unes au-dessus des autres. Lorsque le contenu en question est du texte, on obtient des paragraphes².

La commande de base pour créer une boîte verticale est :

```
\parbox[⟨pos⟩]{⟨largeur⟩}{⟨texte⟩}
```

Ici, l'argument optionnel *⟨pos⟩* permet d'ajuster l'alignement vertical de la boîte avec la ligne de base : **b** ou **t** selon que l'on souhaite aligner, respectivement, le bas ou le haut de la boîte avec la ligne de base. Par défaut, la boîte est centrée avec la ligne de base. Cet argument n'a aucun effet si la boîte est le seul élément de contenu du paragraphe.

On remarquera que l'argument *⟨largeur⟩* est ici obligatoire. Autrement dit, on doit nécessairement définir la largeur des boîtes verticales, un peu

2. D'où l'appellation de *paragraph boxes* en anglais ou *parboxes* dans le jargon \LaTeX .

comme il faut bien définir la largeur de la page pour le texte normal (la classe se charge de ce détail).

Les boîtes créées avec `\parbox` ne peuvent contenir de structures « complexes » comme des listes ou des tableaux. Parce que plus général, l'outil véritablement utile pour la création de boîtes verticales est l'environnement `minipage`. Cet environnement peut contenir à peu près n'importe quel type de contenu. Comme son nom l'indique, c'est ni plus ni moins qu'une page miniature à l'intérieur de la page standard.

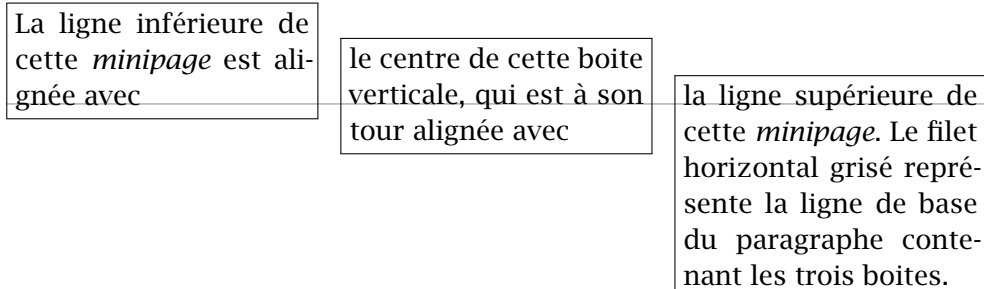
La syntaxe de l'environnement `minipage` est la suivante :

```
\begin{minipage}[\langle pos \rangle]{\langle largeur \rangle}
  \langle texte \rangle
\end{minipage}
```

La signification des arguments `\langle largeur \rangle` et `\langle pos \rangle` est la même que pour la commande `parbox`.

L'environnement `minipage` est fréquemment utilisé pour disposer des éléments de contenu de manière spécifique sur la page, notamment des tableaux ou des figures côte à côte ou en grille (voir l'[exemple 6.8](#) à la [page 77](#)).

Exemple 5.1. L'agencement de boîtes ci-dessous est produit avec le code qui suit immédiatement.



```
\begin{minipage}[b]{0.3\textwidth}
  La ligne inférieure de cette \emph{minipage} [...]
\end{minipage}
\hfill
\parbox{0.3\textwidth}{le centre de cette boîte [...] }
\hfill
\begin{minipage}[t]{0.3\textwidth}
  la ligne supérieure de cette \emph{minipage}. [...]
\end{minipage}
```

□

La commande `\hfill` utilisée entre les boîtes dans l'exemple ci-dessus indique à \TeX d'insérer de l'espace entre les éléments de contenu de manière à remplir entièrement la ligne de texte. C'est une commande très utile pour disposer automatiquement des éléments à intervalles égaux sur la largeur du bloc de texte. Ainsi,

```
\framebox[\linewidth]{gauche \hfill droite}
```

produit

gauche	\hfill	droite
--------	----------	--------

alors que

```
\framebox[\linewidth]{gauche \hfill centre \hfill droite.}
```

produit

gauche	centre	droite.
--------	--------	---------

5.3 Boîtes de réglure

En imprimerie, une réglure est une ligne droite continue ou pointillée. Une ligne n'étant jamais rien d'autre qu'un rectangle plein, si mince fut-il, la réglure est le troisième type de boîte³ dans \TeX .

La commande

```
\rule[\langle\text{déplacement}\rangle]{\langle\text{largeur}\rangle}{\langle\text{hauteur}\rangle}
```

crée une réglure de dimensions $\langle\text{largeur}\rangle \times \langle\text{hauteur}\rangle$. Par défaut, la réglure s'appuie sur la ligne de base. Le résultat de

```
\rule{1cm}{6pt}
```

est donc une ligne pleine de 1 cm de long et de 6 points d'épais : ████████.

L'argument optionnel $\langle\text{déplacement}\rangle$ permet de déplacer verticalement la réglure au-dessus ou au-dessous de la ligne de base selon que la longueur $\langle\text{déplacement}\rangle$ est positive ou négative. Les commandes

```
\rule[3pt]{1cm}{6pt}  
\rule[-3pt]{1cm}{6pt}
```

créent respectivement les réglures ████████ et ████████.

Un usage intéressant de la réglure consiste à faire croire à \TeX qu'une ligne est plus haute qu'il n'y paraît en insérant dans celle-ci une réglure de

3. *Rule box*, en anglais

largeur nulle. Par exemple, la distance entre la présente ligne et les autres du paragraphe est plus grande que la normale parce que j'y ai inséré une réglure invisible avec

```
\rule[-12pt]{0mm}{30pt}
```

Ce truc est particulièrement utile pour augmenter la hauteur des lignes dans un tableau; voir la [section 6.2](#).

5.4 Exercices

Utiliser comme canevas le fichier `exercice-gabarit.tex` pour tous les exercices ci-dessous.

5.1 Une fois qu'une boîte est définie, \TeX n'y voit qu'une unité de contenu avec ses dimensions propres. Il est donc possible de définir une boîte à l'intérieur d'une autre, et ce, peu importe le type de boîte.

Avec ceci en tête, définir la boîte suivante :

Ce bloc de texte est une boîte verticale de 10 cm de large, doublement encadrée et centrée sur la ligne.

5.2 Réaliser l'agencement de boîtes verticales suivant :

<p>Deux boîtes verticales de hauteurs différentes placées côte à côte</p>	<p>alignées sur leurs premières lignes et le bas de la boîte la plus haute alignée sur la ligne de base (représentée ici par le filet horizontal grisé).</p>
---	--

La solution intuitive serait la suivante :

```
\begin{minipage}[b]{...}
  \parbox[t]{...}{...} \hfill \parbox[t]{...}{...}
\end{minipage}
```

Cependant, cette solution produit le résultat suivant (les boîtes sont rendues visibles par des cadres) :

<p>Les deux boîtes sont correctement alignées l'une par rapport à l'autre</p>	<p>mais l'alignement avec la ligne de base est incorrect.</p>
---	---

La raison : pour $\text{T}_{\text{E}}\text{X}$, la `minipage` externe ne contient que deux « caractères » sur une seule ligne de « texte ». La `minipage` est donc correctement alignée sur sa ligne du bas, mais celle-ci se trouve aussi être la ligne du haut.

Pour parvenir au résultat escompté, utiliser la commande `\mbox` pour créer une seconde ligne (vide) dans la `minipage` externe.

5.3 Réaliser l'agencement de boîtes verticales ci-dessous. (La taille de la police est `\footnotesize`.)

La première ligne de cette *parbox* de 30 mm de large est alignée avec celle de la boîte voisine.

Cette *parbox* de 45 mm de large est positionnée de telle sorte que sa première ligne soit alignée avec le haut de la boîte à gauche et la dernière avec le bas de la boîte à droite. La solution intuitive consistant à placer côte à côte trois boîtes avec des arguments de positionnement `t`, `t` et `b` ne fonctionne pas.

Pour parvenir à cette disposition, il faut avoir recours à des lignes invisibles comme dans l'exercice précédent.

La troisième boîte fait 35 mm de large et l'espace entre les boîtes, 5 mm.

6 Tableaux et figures

Les tableaux et graphiques ne sont pas les éléments de texte les plus simples et rapides à créer avec \LaTeX . À ce chapitre, les traitements de texte brillent avec leurs interfaces graphiques permettant de composer un tableau ou un graphique simple pièce par pièce avec la souris.

En revanche, pour ce type de contenu comme pour tout autre, \LaTeX fait ce qu'on lui demande, sans tenter de deviner notre pensée ou, pire, de prétendre savoir mieux que nous ce que nous voulons faire. À ce chapitre, les traitements de texte ne brillent plus! Si vous avez déjà éprouvé de la difficulté à contrôler les bordures d'un tableau, la hauteur des lignes ou la largeur des colonnes dans un traitement de texte, vous savez combien l'exercice de composition d'un tableau avec ces outils peut rapidement devenir frustrant.

Avant de discuter de la création ou de l'insertion de tableaux, de graphiques et d'images dans un document \LaTeX , il convient de présenter très succinctement quelques règles à suivre pour concevoir des tableaux clairs et faciles à consulter.

6.1 De la conception de beaux tableaux

Les tableaux servent à disposer de l'information sous forme de grille. Par conséquent, le premier réflexe pour les mettre en forme consiste souvent à mettre en évidence cette grille par le biais de filets¹ horizontaux et verticaux.

C'est une mauvaise idée, une pratique à éviter. Vraiment!

Comparez les deux tableaux ci-dessous. Le premier est mis en forme selon une approche classique supportée depuis toujours par \LaTeX : filets doubles en entête et en pied de tableau, filets simples entre chaque ligne et entre les colonnes.

1. Terme typographique pour ce qui est communément appelés des « lignes » dans le langage courant ou des « bordures » dans les logiciels de traitement de texte. Dans la documentation en anglais, on parle de *rules*.

i	v	b_i	$\lfloor v/b_i \rfloor$	$v \bmod b_i$	x_i
0	91 492	60	1 524	52	52
1	1 524	60	25	24	24
2	25	24	1	1	1
3	1	365	0	1	1

Le second tableau tire profit des fonctionnalités du paquetage **booktabs** (Fear, 2020) et des recommandations de son auteur : les filets horizontaux sont d'épaisseur différente selon qu'ils sont situés dans l'entête et dans le pied du tableau ou entre les lignes, l'espace autour des filets horizontaux est plus grand et, surtout, il n'y a pas de filets verticaux.

i	v	b_i	$\lfloor v/b_i \rfloor$	$v \bmod b_i$	x_i
0	91 492	60	1 524	52	52
1	1 524	60	25	24	24
2	25	24	1	1	1
3	1	365	0	1	1

La seconde version n'est-elle pas la plus aérée et la plus facile à consulter ? N'est-ce pas que, contrairement à ce que l'on pourrait penser, les filets verticaux ne sont pas du tout requis pour bien délimiter les colonnes ?

booktabs

Tel que mentionné ci-dessus, le paquetage **booktabs** ajoute des fonctionnalités à \LaTeX pour améliorer la qualité typographique des tableaux. Dans la [documentation](#) du paquetage, son auteur énonce quelques règles à suivre pour la mise en forme des tableaux :

1. Ne *jamais* utiliser de filets verticaux. Si l'information du côté gauche du tableau semble si différente de celle du côté droit qu'un filet vertical apparaît absolument nécessaire, scinder simplement l'information dans deux tableaux ;
2. Ne jamais utiliser de filets doubles ;
3. Placer les unités (\$, cm, °C, etc.) dans le titre de la colonne plutôt qu'après chaque valeur dans le corps du tableau ;
4. Toujours inscrire un chiffre du côté gauche du séparateur décimal : 0,1 et non ,1 (pratique plus répandue en anglais, où le séparateur décimal est le point) ;
5. Ne pas utiliser un symbole pour représenter une valeur répétée (comme " ou —). Laisser un blanc ou répéter la valeur s'il subsiste une ambiguïté.

Je recommande évidemment de suivre ces règles et c'est pourquoi le présent ouvrage ainsi que les fichiers d'exemples font usage des commandes de **booktabs**.

Les fonctionnalités de **booktabs** sont intégrées à la classe memoir. Il n'est donc pas nécessaire de charger le paquetage si vous utilisez cette classe.

6.2 Tableaux

Peu importe l'outil informatique utilisé, la création d'un tableau requiert toujours de préciser à l'ordinateur le nombre de colonnes que contiendra le tableau, l'entête du tableau, le cas échéant, et le contenu des différentes cellules. Cette dernière étape nécessite à son tour une convention pour indiquer les passages à la colonne suivante ainsi que le passage à la ligne suivante.

On crée des tableaux dans \LaTeX principalement avec les environnements `tabular`, `tabular*` et `tabularx` (ce dernier fourni par le paquetage **tabularx** ou par la classe memoir). La syntaxe de ces environnements est :

```
\begin{tabular}{\langle format \rangle} \langle lignes \rangle \end{tabular}
\begin{tabular*}{\langle largeur \rangle}{\langle format \rangle} \langle lignes \rangle \end{tabular*}
\begin{tabularx}{\langle largeur \rangle}{\langle format \rangle} \langle lignes \rangle \end{tabularx}
```

La signification des arguments ² est la suivante. Je ne traite ici que les options les plus souvent utilisées. Pour une liste plus exhaustive, consulter la [documentation](#) de la classe memoir (chapitre 11) ou [Wikilivres](#) (2023, section [Tableaux](#)).

memoir

`\langle largeur \rangle` Largeur hors tout d'un tableau avec les environnements `tabular*` et `tabularx`. Dans l'environnement `tabular`, la largeur d'un tableau est déterminée automatiquement pour contenir tout le tableau, quitte à dépasser dans la marge de droite.

La largeur du tableau est généralement exprimée en fraction de la largeur du bloc de texte (longueur `\textwidth`). Par exemple, les déclarations suivantes définissent respectivement des tableaux occupant toute la largeur d'une page et 80 % de la largeur de la page :

```
\begin{tabular*}{\textwidth}{\langle format \rangle}
```

```
\begin{tabularx}{0.8\textwidth}{\langle format \rangle}
```

2. J'ai omis un argument optionnel rarement utilisé servant à spécifier l'alignement vertical du tableau par rapport à la ligne de base externe.

L'environnement `tabular*` joue sur l'espace entre les colonnes pour parvenir à la largeur prescrite, alors que `tabularx` joue sur la largeur des colonnes (voir ci-dessous).

format) Le format des colonnes et, par le fait même, le nombre de colonnes puisque l'argument doit compter un symbole pour chaque colonne du tableau. Les principaux symboles de mise en forme des colonnes sont :

`l` contenu de la colonne aligné à gauche ;

`r` contenu de la colonne aligné à droite ;

`c` contenu de la colonne centré ;

`p{< lgr >}` contenu de la colonne traité comme un paragraphe de texte de largeur *lgr* ;

`X` [environnement `tabularx` seulement] colonne dont la largeur peut être ajustée pour obtenir un tableau de la largeur prescrite ; identique à `p` par ailleurs.

Par exemple, la déclaration

```
\begin{tabular}{lrp{5cm}}
```

définit un tableau à trois colonnes dont le contenu de la première est aligné à gauche ; celui de la deuxième est aligné à droite ; celui de la troisième est en texte libre dans une cellule d'une largeur de 5 cm.

Avec la déclaration

```
\begin{tabularx}{\textwidth}{lrX}
```

la largeur de la troisième colonne sera plutôt adaptée automatiquement pour que le tableau occupe toute la largeur de la page.

Les symboles `|` et `||` dans *format* servent à insérer des filets verticaux simples et doubles entre les colonnes, mais nous avons vu à la [section 6.1](#) que c'est une pratique à proscrire.

lignes) Le contenu des cellules du tableau. Les entrées des cellules sont séparées par le symbole « `&` » et les lignes par « `\\` ». Une cellule peut être vide.

Outre du texte, les lignes de contenu peuvent contenir certaines commandes spéciales pour contrôler la mise en forme du tableau. En premier lieu, la commande

```
\multicolumn{<n>}{<fmt>}{<texte>}
```

permet de fusionner les $\langle n \rangle$ cellules suivantes en une seule de format $\langle fmt \rangle$ et d'y placer $\langle texte \rangle$. Cette commande ne peut apparaître qu'au début d'une ligne ou après un symbole de changement de colonne « & ». Elle est souvent utilisée avec une valeur de $\langle n \rangle$ égale à 1 pour changer le format d'une cellule, par exemple pour centrer le titre d'une colonne qui est autrement alignée à gauche ou à droite.

Ensuite, les commandes suivantes³ servent à insérer des filets horizontaux dans un tableau :

```
\toprule
\midrule
\cmidrule{\langle m-n \rangle}
\bottomrule
```

La commande `\toprule` insère un filet horizontal épais suivi d'un espace vertical au début d'un tableau; `\midrule` insère un filet horizontal mince précédé et suivi d'un espace vertical entre deux lignes; `\cmidrule{\langle m-n \rangle}` insère un filet horizontal comme `\midrule` de la gauche de la colonne $\langle m \rangle$ à la droite de la colonne $\langle n \rangle$; enfin, `\bottomrule` insère un filet horizontal épais précédé d'un espace vertical à la fin d'un tableau. Une fin de ligne « `\` » doit obligatoirement précéder chacune de ces commandes, sauf évidemment `\toprule`.

La hauteur des lignes d'un tableau est déterminée automatiquement en fonction du contenu de celles-ci.

Exemple 6.1. Dans le tableau ci-dessous, la première colonne est alignée à gauche et toutes les autres sont alignées à droite. La largeur totale du tableau est ajustée à son contenu.

Produit	Quantité	Prix unitaire (\$)	Prix (\$)
Vis à bois	2	9,90	19,80
Clous vrillés	5	4,35	21,75
TOTAL	7		41,55

Le code ci-dessous permet de créer ce tableau. Observez comment les lignes de contenu sont définies.

```
\begin{tabular}{lrrr}
\toprule
```

3. Ce sont les commandes de **booktabs** et memoir auxquelles je faisais référence à la [section 6.1](#).

```

Produit & Quantité & Prix unitaire (\$) & Prix (\$) \\
\midrule
Vis à bois & 2 & 9,90 & 19,80 \\
Clous vrillés & 5 & 4,35 & 21,75 \\
\midrule
TOTAL & 7 & & 41,55 \\
\bottomrule
\end{tabular}

```

□

Exemple 6.2. Modifions le tableau de l'[exemple 6.1](#) pour obtenir le tableau suivant :

Produit	Quantité	Prix unitaire (\$)	Prix (\$)
Vis à bois	2	9,90	19,80
Clous vrillés	5	4,35	21,75
TOTAL	7		41,55

Le tableau occupe désormais toute la largeur de la page, la largeur de la première colonne étant ajustée pour combler l'espace nécessaire. De plus, le titre de la première colonne est centré et la hauteur de l'entête est augmentée.

Le code suivant permet de réaliser cette mise en forme.

```

\begin{tabularx}{\textwidth}{Xrrr}
\toprule
\multicolumn{1}{c}{Produit} &
\rule[-8pt]{0mm}{24pt} Quantité &
Prix unitaire (\$) & Prix (\$) \\
\midrule
Vis à bois & 2 & 9,90 & 19,80 \\
Clous vrillés & 5 & 4,35 & 21,75 \\
\midrule
TOTAL & 7 & & 41,55 \\
\bottomrule
\end{tabularx}

```

L'environnement `tabularx` sert à créer un tableau de largeur définie et la commande `\multicolumn`, à centrer le titre de la première colonne. La hau-

teur de l'entête a été augmentée à l'aide d'une réglure invisible (section 5.3) insérée quelque part sur la ligne. □

6.3 Figures et graphiques

Il est possible de tracer des figures simples directement avec \LaTeX . Par « simple » j'entends : des figures se limitant pour l'essentiel à du texte, des lignes, des flèches, des ronds et des ovales. C'est parfois amplement suffisant et, en définitive, c'est assez pratique puisque le code source d'une figure se trouve alors dans le même format que le reste du document.

Pour créer des figures et des graphiques plus complexes, on a généralement recours à des logiciels spécialisés externes. \LaTeX est ensuite en mesure d'importer des graphiques dans les formats standards tels que PDF, JPEG ou PNG, voire même d'insérer dans un document une ou plusieurs pages d'un document PDF.

Couvrir les détails de la création et de la manipulation d'images dépasse largement la portée du présent ouvrage. Le reste de cette section ne présente que les principales fonctionnalités.

6.3.1 Figures \LaTeX

L'environnement `picture` permet de tracer des figures simples comme des diagrammes à base de texte, des flux logiques ou des organigrammes. Quelques logiciels spécialisés de création de graphiques sont même en mesure d'exporter leurs graphiques dans le format de `picture`.

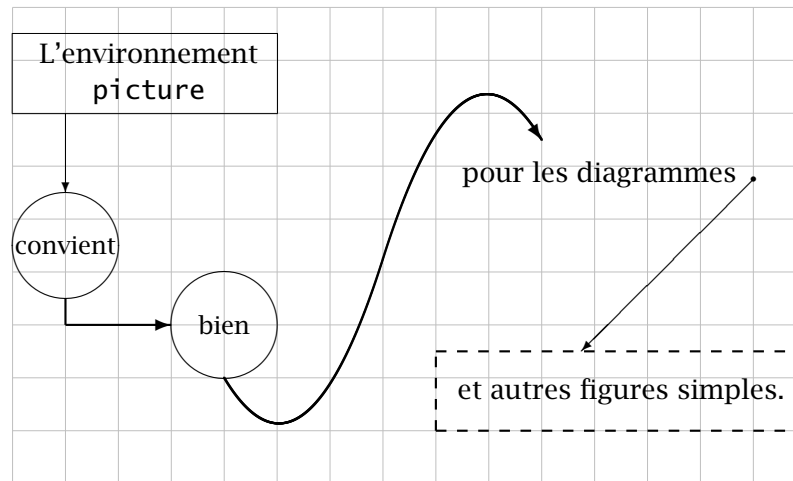
Une fois conçues, les figures réalisées avec `picture` sont simples à modifier ; nul besoin de recourir à un logiciel externe pour le moindre petit changement. Autre avantage : la police du texte de la figure sera le même que celle du document.

Le tracage d'une figure avec l'environnement `picture` requiert d'abord une grille (invisible) d'une dimension quelconque dans l'unité de mesure de votre choix (autrement dit : les lignes de la grille peuvent être distantes aussi bien de 1pt que de 1cm). Ensuite, vous disposez des éléments sur la grille en donnant les coordonnées du point d'ancrage et, le cas échéant, les dimensions de l'élément, la distance à parcourir ou quelque autre information pour compléter l'élément. C'est souvent plus simple d'esquisser d'abord un modèle au crayon sur du papier quadrillé.

La figure ci-dessous illustre ce qu'il est possible de faire avec l'environnement `picture`. La consultation du code commenté correspondant devrait vous permettre de comprendre les principes de base de la création de figures.

📖 `memoir` Autrement, l'annexe D de la [documentation](#) de `memoir` fournit une bonne introduction à `picture`.

(J'ai tracé la grille en filigrane dans la figure afin de faciliter la comparaison entre le code et le résultat.)



```

\setlength{\unitlength}{7mm} % unité de mesure
\begin{picture}(15,9) % grille 15 x 9
%%%
%%% On trace d'abord toutes les boites
%%%
%% Rectangle "L'environnement picture"
\put(0,7){% % point d'ancrage (0, 7)
  \framebox(5,1.5){% % rectangle 5 x 1,5 plein
    \begin{minipage}{35mm} % contenu de la boite
      \centering L'environnement \ \ \texttt{picture}
    \end{minipage}}}}

%% Cercles "convient" et "bien"
\put(1,4.5){\circle{2}} % cercle diamètre 2
\put(1,4.5){\makebox(0,0){\small convient}} % texte centré
\put(4,3){\circle{2}} % autre cercle
\put(4,3){\makebox(0,0){\small bien}} % texte

%% Texte "pour les diagrammes"
\put(8.5,5.7){pour les diagrammes} % point d'ancrage (8,5, 5,7)

%% Rectangle pointillé "et autres figures simples."
\thicklines % lignes grasses

```

```

\put(8,1){\dashbox{0.2}(7,1.5){ % rectangle 7 x 1,5 pointillé
    et autres figures simples.}}

%%%
%%% On trace ensuite les lignes entre les boites
%%%
% De "L'environnement picture" à "convient"
\thinlines % retour aux lignes minces
\put(1,7){\vector(0,-1){1.5}} % flèche vers le bas longueur 1,5
% [couple (0,-1) donne la pente]

% De "pour les diagrammes" à "et autres figures simples."
\put(14,5.75){\circle*{0.1}} % petit cercle plein
\put(14,5.75){\vector(-1,-1){3.25}} % flèche vers sud-ouest
% [3.25 = déplacement hor.]

% Entre les deux cercles; requiert deux segments
\thicklines % lignes grasses
\put(1,3.5){\line(0,-1){0.5}} % courte ligne vert. sans flèche
\put(1,3){\vector(1,0){2}} % flèche horizontale

% Entre "bien" et "pour les diagrammes"; requiert deux courbes
% de Bézier placées bout à bout pour produire une courbe en S
\qbezier(4,2)(5.5,-0.5)(7,4.25) % bas du S
\qbezier(7,4.25)(8.5,9)(10,6.5) % haut du S
\put(10,6.5){\vector(2,-3){0}} % pointe de flèche seule
\end{picture}

```

Il existe quelques outils pour tracer des figures plus complexes directement avec \TeX , dont PSTricks (Van Zandt et collab., 2014) ou le système TikZ/PGF (Tantau, 2023). Ce dernier est maintenant assez populaire pour aspirer au statut d'outil standard dans le monde de \TeX .

6.3.2 Importation d'images

Il est aujourd'hui simple d'importer des images de source externes dans un document \TeX en utilisant l'un ou l'autre des paquetages **graphics** ou **graphicx** (Carlisle et The \TeX 3 Project, 2021) en combinaison avec un moteur \TeX moderne tel que pdf \TeX ou Xe \TeX . Les fonctionnalités des deux paquetages sont les mêmes, seules les syntaxes des commandes diffèrent. Je présente les commandes de **graphicx**, plus modernes et conviviales.

La commande de base pour importer des images dans un document \TeX est

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

où *fichier* est le nom du fichier à importer. Il n'est pas nécessaire de préciser l'extension dans le nom de fichier pour les types d'images usuelles. Avec les moteurs pdf \TeX et Xe \TeX , les types d'images automatiquement reconnus sont au moins PDF, JPEG et EPS.

Les *options* de `\includegraphics`, nombreuses, permettent de redimensionner une image, de la faire pivoter ou encore de n'en importer qu'une partie. L'exemple ci-dessous présente les principales fonctionnalités ; consulter la [documentation](#) pour les détails et d'autres options.

graphics

Exemple 6.3. Le fichier `ul_p.pdf` contenant le logo de l'Université Laval en couleur et en format vectoriel est distribué avec le présent ouvrage. La simple commande

```
\includegraphics{ul_p}
```

insère le fichier en pleine grandeur dans le document :



On peut redimensionner l'image en valeur relative avec l'option `scale` ou en valeur absolue avec les options `width` ou `height` :

```
% réduction à 40 % de taille réelle
\includegraphics[scale=0.4]{ul_p}
```



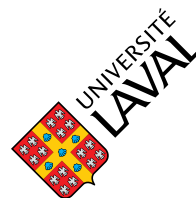
```
% réduction à 15 mm de haut
\includegraphics[height=15mm]{ul_p}
```



(Il est préférable d'utiliser une seule de `width` ou `height`. Autrement, ajoutez l'option `keepaspectratio=true` pour éviter de déformer l'image.)

L'option `angle` permet de faire pivoter l'image dans le sens inverse des aiguilles d'une montre autour du coin inférieur gauche de l'image :

```
%% réduction à 25 %, rotation à 45 degrés
\includegraphics[angle=45,scale=0.25]{u1_p}
```



Enfin, il y a diverses manières de sélectionner une partie seulement d'une image. L'option `bb` (pour *Bounding Box*) prend quatre mesures en points PostScript (tableau 2.1) définissant le coin inférieur gauche et le coin supérieur droit de la zone à inclure :

```
%% extraction du logo seul et réduction
\includegraphics[bb=0 0 102 129,clip=true,
scale=0.4]{u1_p}
```



□

La commande `\includegraphics` permet d'appliquer certaines transformations aux images importées. Ces transformations peuvent également s'effectuer à l'aide de commandes externes *après* l'importation. L'avantage de ces commandes, c'est qu'elles sont valides tout autant pour du texte que pour des images.

Le paquetage **graphicx** définit les commandes suivantes :

```
\rotatebox[options]{angle}{texte}
\scalebox{échelle-h}[échelle-v]{texte}
\resizebox{dim-h}{dim-v}{texte}
\reflectbox{texte}
```

Dans tous les cas, `<texte>` peut être du simple texte ou une boîte quelconque, y compris le résultat de `\includegraphics`. Ainsi,

```
\rotatebox{45}{\includegraphics{u1_p}}
```

et

```
\includegraphics[angle=45]{u1_p}
```

donnent le même résultat.

Avec `\scalebox`, la mise à l'échelle `<échelle-h>` s'applique par défaut autant à l'horizontale qu'à la verticale. Autrement, `<texte>` est déformé. Avec

`\resizebox`, on peut spécifier l'une de $\langle dim-h \rangle$ ou $\langle dim-v \rangle$ et `!` pour l'autre valeur pour éviter de déformer $\langle texte \rangle$.

Exemple 6.4. Voici des exemples d'utilisation des commandes `\rotatebox`, `\scalebox`, `\resizebox` et `\reflectbox` avec du texte :

<code>\rotatebox{135}{texte}</code>	texte
<code>\scalebox{1.5}{texte}</code>	texte
<code>\scalebox{1.5}[0.75]{texte}</code>	texte
<code>\resizebox{3cm}{!}{texte}</code>	texte
<code>\reflectbox{texte}</code>	texte


□

6.3.3 Insertion de documents PDF

Il est parfois utile d'insérer dans un document \LaTeX une ou plusieurs pages d'un autre document en format PDF, et ce, sans avoir à se soucier des marges respectives des deux documents. Si l'on utilise les moteurs pdf \LaTeX ou Xe \LaTeX , le très pratique paquetage **pdfpages** (Matthias, 2022) fournit la commande

```
\includepdf[\langle options \rangle]{\langle fichier \rangle}
```

■ pdfpages

Les $\langle options \rangle$ sont très nombreuses ; consultez la [documentation](#) .

Exemple 6.5. Il n'est pas rare que les couvertures avant et arrière d'un document soient réalisées dans un logiciel spécialisé de création graphique. Supposons que les deux couvertures sont sauvegardées en format PDF dans un fichier `couvertures.pdf`. Pour les utiliser dans le document, il suffit de placer aux endroits appropriés les commandes

```
\includepdf[pages=1]{couvertures}
\includepdf[pages=2]{couvertures}
```

□

6.4 Éléments flottants

Dans la terminologie de \LaTeX , un élément flottant⁴ est un bloc de contenu (une boîte, en fait) que le logiciel pourra positionner sur la page et dans le document plus ou moins automatiquement en fonction d'un algorithme prédéfini. C'est une fonctionnalité très évoluée de \LaTeX .

Pourquoi voudriez-vous laisser \LaTeX décider où un élément de contenu devrait se retrouver dans votre document ? D'abord et avant tout pour les tableaux et les figures. En effet, les tableaux et les figures occupent souvent beaucoup d'espace vertical dans la page. S'il ne reste plus assez de place pour y afficher un tel élément de contenu, \LaTeX devra le déplacer au début de la page suivante et cela risque de produire une page inesthétique car insuffisamment remplie⁵. Les traitements de texte génèrent sans rechigner des pages à demi remplies dans de telles situations.

Définir un élément comme flottant laisse plutôt à \LaTeX la possibilité de le disposer au meilleur endroit en fonction de la taille de l'élément, du contenu du document et de diverses règles typographiques.

Les environnements `table` et `figure` permettent de créer des éléments flottants :

```
\begin{table}[\langle pos \rangle] \langle tableau \rangle \end{table}
\begin{figure}[\langle pos \rangle] \langle figure \rangle \end{figure}
```

Ci-dessus, `\langle tableau \rangle` et `\langle figure \rangle` représentent le code source d'un tableau ou d'une figure avec possiblement une commande `caption`, tel que traité plus loin.

L'argument optionnel `\langle pos \rangle` permet d'indiquer à \LaTeX la ou les positions *souhaitées* pour le tableau ou la figure dans la page. Lorsqu'il est question d'éléments flottants, il est très difficile de donner des ordres fermes à \LaTeX et l'effet de l'argument `\langle pos \rangle` est souvent déconcertant. Aussi vaut-il souvent mieux ne rien indiquer et laisser \LaTeX faire à sa guise. Le résultat demeure assez prévisible puisque \LaTeX tâchera d'insérer l'élément flottant dans le document *dès que possible* sous réserve des conditions suivantes :

- ▶ l'élément flottant ne peut apparaître dans le document avant la page où l'élément est défini ;
- ▶ l'élément sera placé de préférence dans le haut de la page courante, puis dans le bas et enfin sur une page séparée ne pouvant contenir que des éléments flottants, mais pas de texte.

4. *Float* en anglais.

5. *Underful \vbox* dans le jargon de \TeX .

Produit	Quantité	Prix unitaire (\$)	Prix (\$)
Vis à bois	2	9,90	19,80
Clous vrillés	5	4,35	21,75
TOTAL	7		41,55

Si la décision de \LaTeX ne convient pas, il est possible de l'infléchir avec une combinaison d'une ou plusieurs des lettres suivantes dans l'argument $\langle pos \rangle$;

b placer l'élément au bas (*bottom*) de la page ;

h placer l'élément ici (*here*), à l'endroit où il est défini dans le code source ;

p placer l'élément sur une page séparée ;

t placer l'élément au haut (*top*) de la page ;

! essayer plus fort de placer l'élément à l'endroit spécifié dans le reste de l'argument.

📖 memoir

La valeur par défaut de l'argument $\langle pos \rangle$ est `tbp`. La section 10.4 de la [documentation](#) de memoir explique plus en détail la signification des valeurs ci-dessus. Si vous voulez vraiment *tout* savoir sur la disposition des éléments flottants, consultez [Mittelbach \(2014a\)](#).

Exemple 6.6. Reprenons le tableau de l'[exemple 6.1](#), mais cette fois défini à l'intérieur d'un environnement `table` :

```
\begin{table}
  \centering
  \begin{tabular}{lrrr}
    \toprule
    Produit & Quantité & Prix unitaire (\$) & Prix (\$) \\
    \midrule
    Vis à bois & 2 & 9,90 & 19,80 \\
    Clous vrillés & 5 & 4,35 & 21,75 \\
    \midrule
    TOTAL & 7 & & 41,55 \\
    \bottomrule
  \end{tabular}
\end{table}
```

\LaTeX a automatiquement placé le tableau dans le document en fonction des règles précitées. □

```
\addto\captionsfrench{%
  \def\frenchtablename{\scshape Tab.}}
\addto\captionsfrench{%
  \def\frenchfigurename{\scshape Fig.}}
```

FIG. 6.1 – Commandes pour modifier les étiquettes des tableaux et des figures utilisées par **babel**

Dans un document soigné, tout tableau et toute figure devrait comporter une légende ainsi qu'un numéro afin de pouvoir les annoncer et y faire référence dans le texte (« comme l'illustre la figure 3... »). Cela permet à la fois de guider le lecteur au fil de sa lecture et de construire une liste des tableaux et des figures⁶ dans les pages liminaires d'un long document.

Pour ajouter une légende à un tableau ou une figure, il suffit d'utiliser à l'intérieur des environnements `table` et `figure` la commande

```
\caption[<texte_court>]{<texte>}
```

où *<texte>* est le texte de la légende. Si celui-ci est long (plus d'une ligne), on peut en fournir une version abrégée dans l'argument optionnel *<texte_court>*. C'est cette version abrégée qui sera utilisée dans la liste des tableaux ou dans la liste des figures.

La commande `\caption` insère, à l'endroit où elle apparaît dans l'environnement, une légende de la forme « TABLE *n* - *<texte>* » pour un tableau ou « FIGURE *n* - *<texte>* » pour une figure. Le texte de la légende est centré sur la page lorsqu'il fait moins d'une ligne; dans le cas contraire il est disposé comme un paragraphe normal.



Les anciennes versions du style français de **babel** utilisaient les étiquettes plus neutres « TAB. » et « FIG. » dans les légendes des tableaux et figures. Pour utiliser — comme dans le présent document — ces versions plutôt que les nouvelles versions par défaut, ajoutez dans le préambule les commandes de la [figure 6.1](#).

Les références à un tableau ou à une figure dans le texte sont insérées dans un document avec le système de renvois automatiques de \LaTeX ([section 3.4](#)). Pour attribuer une étiquette à l'élément flottant, placez la com-

6. Obtenues respectivement avec les commandes `\listoftables` et `\listoffigures` mentionnées à la [section 3.3](#).

```

\begin{table}
  \centering
  \caption{Tableau correspondant au code
    de la \autoref{fig:[...]}}
  \label{tab:[...]}
  \begin{tabular}{lrrr}
    \toprule
    Produit & Quantité & Prix unitaire (\$) & Prix (\$) \\
    \midrule
    Vis à bois & 2 & 9,90 & 19,80 \\
    Clous vrillés & 5 & 4,35 & 21,75 \\
    \midrule
    TOTAL & 7 & & 41,55 \\
    \bottomrule
  \end{tabular}
\end{table}

```

FIG. 6.2 - Code source pour créer le [tableau 6.1](#)TAB. 6.1 - Tableau correspondant au code de la [figure 6.2](#)

Produit	Quantité	Prix unitaire (\$)	Prix (\$)
Vis à bois	2	9,90	19,80
Clous vrillés	5	4,35	21,75
TOTAL	7		41,55

mande `\label` dans le texte de la commande `\caption` ou dans son voisinage immédiat. Les commandes `\ref` ou `\autoref` servent ensuite à insérer des renvois dans le texte.

Exemple 6.7. Voici la recette complète pour composer un tableau et une figure dans \LaTeX , légende et renvoi inclus. Le code source de la [figure 6.2](#) crée le [tableau 6.1](#). □

Les environnements `table` et `figure` créent des éléments flottants qui, par ailleurs, sont des boîtes verticales standards ([section 5.2](#)). Il est donc permis d'y mettre à peu près n'importe quoi, mais surtout plus d'un tableau ou plus d'une figure (ou même une combinaison des deux). Les environne-

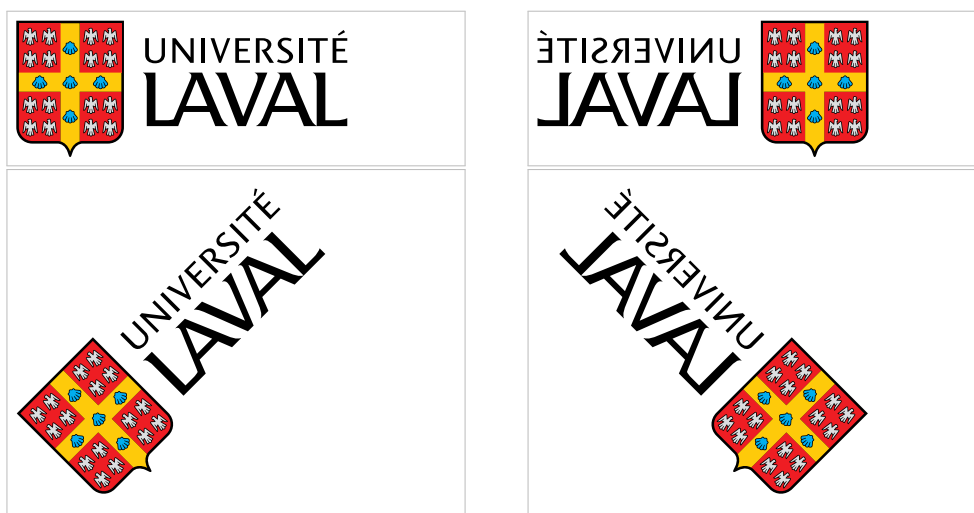


FIG. 6.3 – Exemple de disposition de plusieurs graphiques dans une même figure flottante. Les rectangles en grisé indiquent les limites des boîtes verticales.

ments `minipage` (section 5.2) se révèlent alors particulièrement utiles pour disposer les éléments de contenu dans la boîte.

Exemple 6.8. La figure 6.3 contient quatre images sous forme de grille 2×2 . Le code ci-dessous démontre comment parvenir à cette disposition à l'aide de boîtes verticales créées avec l'environnement `minipage`.

```
\begin{figure}
  \begin{minipage}{0.45\linewidth}
    \includegraphics[scale=0.4]{ul_p}
  \end{minipage}
  \hfill
  \begin{minipage}{0.45\linewidth}
    \reflectbox{\includegraphics[scale=0.4]{ul_p}}
  \end{minipage}
  \newline
  \begin{minipage}{0.45\linewidth}
    \includegraphics[scale=0.4,angle=45]{ul_p}
  \end{minipage}
  \hfill
  \begin{minipage}{0.45\linewidth}
    \reflectbox{\includegraphics[scale=0.4,angle=45]{ul_p}}
  \end{minipage}
\end{figure}
```

```
\end{minipage}
\end{figure}
```

□

📖 `memoir`

Lorsqu'une figure ou un tableau compte plusieurs éléments, comme à l'exemple précédent, il peut être souhaitable d'ajouter pour chacun une sous-légende. L'[exercice 6.3](#) explique comment y parvenir. La section 10.9 de la [documentation](#) de `memoir` comporte de nombreux détails additionnels sur les sous-légendes.

6.5 Exercices

6.1 Reproduire le tableau ci-dessous à l'aide d'un environnement `tabular`. Utiliser le gabarit de document `exercice-gabarit.tex`.

La première colonne est alignée à gauche, la seconde est un bloc de texte de 7,5 cm et la troisième est alignée à droite. Le symbole N° dans l'entête est produit par la commande `\No` de `babel`. Le dernier prix est composé avec la commande `\nombre` de `numprint`.

N° lot	Description	Prix (\$)
U-236	Ordinateur portable MacBook Air 13 pouces mi-2013, processeur 1,3 GHz, 8 Go RAM, disque SSD 250 Go	998
U-374	Chaise de bureau ergonomique ajustable de 8 façons, revêtement de tissu gris foncé	275
U-588	Table de travail en L	1 125

6.2 Apporter au tableau de l'exercice précédent les modifications suivantes : centrer le titre de la deuxième colonne ; ajuster automatiquement la largeur du tableau au bloc de texte sur la page avec un environnement `tabularx`.

6.3 L'[exemple 6.8](#) montre comment intégrer plusieurs figures (ou tableaux) à l'intérieur d'un même environnement flottant en les disposant dans des boîtes verticales. Dans de tels cas, il peut être souhaitable de fournir une légende pour l'ensemble du flottant, mais aussi des sous-légendes pour chaque tableau ou figure.

Avec la classe `memoir`, la production de sous-légendes requiert d'abord de déclarer, dans le préambule du document, son intention d'en créer

pour les environnements flottants `table` ou `figure` avec, selon le cas, les commandes

```
\newsubfloat{table}  
\newsubfloat{figure}
```

Ensuite, on utilise la commande

```
\subcaption{\langle texte \rangle}
```

de la même manière que `\caption`.

Le fichier `exercice-subcaption.tex` contient la structure de base pour composer deux tableaux côte à côte. Ajouter des sous-légendes à l'intérieur de l'environnement flottant.

- 6.4 Utiliser le fichier `exercice-gabarit.tex` pour composer un document qui insère, disons, la page couverture du présent document à l'aide des fonctionnalités du paquetage **pdfpages** décrites à la [section 6.3.3](#).
- 6.5 Le document `exercice-demo.tex` contient plusieurs éléments flottants, tableaux et figures. Examiner le code et modifier l'argument optionnel de position d'un flottant pour voir son effet sur la mise en page du document.

7 Mathématiques

S'il est un domaine où \LaTeX brille particulièrement, c'est bien dans la préparation et la présentation d'équations mathématiques — des plus simples aux plus complexes. Après tout, l'amélioration de la qualité typographique des équations mathématiques dans son ouvrage phare *The Art of Computer Programming* figurait parmi les objectifs premiers de Knuth lorsqu'il a développé \TeX . Aujourd'hui, le langage développé par Knuth pour décrire les équations mathématiques est devenu une sorte de standard *de facto*.

7.1 Principes de base du mode mathématique

La mise en forme d'équations mathématiques requiert d'indiquer à l'ordinateur, dans un langage spécial, le contenu de celles-ci et la position des symboles : en exposant, en indice, sous forme de fraction, etc. L'ordinateur peut ensuite assembler le tout à partir de règles typographiques portant, par exemple, sur la représentation des variables et des constantes, l'espacement entre les symboles ou la disposition des équations selon qu'elles apparaissent au fil du texte ou hors d'un paragraphe.

On indique à \LaTeX que l'on change de « langage », par l'utilisation d'un mode mathématique. Il y a deux grandes manières d'activer le mode mathématique :

1. en insérant le code entre les symboles $\$ \$$ pour générer une équation « en ligne », ou au fil du texte ;

on sait que $\$(a + b)^2 = a^2 + 2ab + b^2\$, d'où on obtient...$

on sait que $(a + b)^2 = a^2 + 2ab + b^2$, d'où on obtient...

2. en utilisant un environnement servant à créer une équation hors paragraphe ;

```
on sait que
\begin{equation*}
(a + b)^2
= a^2 + 2ab + b^2,
\end{equation*}
d'où on obtient...
```

on sait que
 $(a + b)^2 = a^2 + 2ab + b^2,$
d'où on obtient...

Dans l'exemple ci-dessus, l'environnement `equation*` (tiré du paquetage **amsmath**, voir la section suivante) crée une équation hors paragraphe, centrée sur la ligne et non numérotée. Avec l'environnement `equation` (donc sans `*` dans le nom), \LaTeX ajoute automatiquement un numéro d'équation séquentiel aligné sur la marge de droite :

```
on sait que
\begin{equation}
(a + b)^2 = a^2 + b^2,
\end{equation}
d'où on obtient...
```

on sait que
 $(a + b)^2 = a^2 + b^2, \quad (7.1)$
d'où on obtient...

Cette disposition est la plus usuelle dans les ouvrages mathématiques. Le type de numérotation diffère selon qu'un document comporte des chapitres ou non.

En mode mathématique, les chiffres sont automatiquement considérés comme des constantes, les lettres comme des variables et une suite de lettres comme un produit de variables (nous verrons plus loin comment représenter des fonctions mathématiques comme `sin`, `log` ou `lim`). Ceci a trois conséquences principales :

1. conformément aux conventions typographiques, les chiffres sont représentés en caractère romain et les variables, en italique;

```
| $123xyz$      123xyz
```

2. l'espace entre les constantes, les variables et les opérateurs mathématiques est géré automatiquement;

```
| $z = 2 x + 3 x y$      z = 2x + 3xy
```

3. les espaces dans le code source n'ont aucun impact sur la disposition d'une équation.

`$z=2x + 3xy$` $z = 2x + 3xy$


Quant au langage retenu par \LaTeX pour décrire les équations mathématiques, il est très similaire à celui que l'on utiliserait pour le faire à voix haute. Il faut simplement recourir à des commandes pour identifier les symboles mathématiques que l'on ne retrouve pas sur un clavier usuel, comme les lettres grecques, les opérateurs d'inégalité ou les symboles de sommes et d'intégrales.

7.2 Un paquetage incontournable

Le paquetage **amsmath** ([American Mathematical Society, 2002](#)) produit par la *American Mathematical Society* fournit diverses extensions à \LaTeX pour faciliter encore davantage la saisie d'équations mathématiques complexes et en améliorer la présentation. L'utilisation de ce paquetage doit être considérée incontournable pour tout document contenant plus que quelques équations très simples.

Au chapitre des améliorations fournies par **amsmath**, notons particulièrement :

- ▶ plusieurs environnements pour les équations hors paragraphe, en particulier pour les équations multilignes ;
- ▶ une meilleure gestion de l'espacement autour des symboles de relation (comme les signes d'égalité) dans les équations multilignes ;
- ▶ une commande pour faciliter l'entrée de texte à l'intérieur du mode mathématique ;
- ▶ un environnement pour la saisie des matrices et des coefficients binomiaux ;
- ▶ des commandes pour les intégrales multiples ;
- ▶ la possibilité de définir de nouveaux opérateurs mathématiques.

Je décris certaines de ces fonctionnalités dans la suite, mais tout utilisateur du paquetage devrait impérativement consulter sa [documentation complète](#) .

 amsmath

7.3 Principaux éléments du mode mathématique

Cette section explique comment créer et assembler divers éléments d'une formule mathématique : exposants, indices, fractions, texte, etc. Les seuls

symboles utilisés sont pour le moment les chiffres et les lettres latines. La présentation d'une partie de l'éventail de symboles mathématiques offerts par \LaTeX fera l'objet de la [section 7.4](#).

7.3.1 Exposants et indices

\LaTeX permet de créer facilement et avec la bonne taille de symboles n'importe quelle combinaison d'exposants et d'indices.

On place un caractère en ^{exposant} avec la commande `^` et en _{indice} avec la commande `_`. Les indices et exposants se combinent naturellement.

`| x^2` x^2 `| a_n` a_n `| x_i^k` x_i^k

(L'ordre de saisie n'a pas d'importance; le troisième exemple donnerait le même résultat avec `x^k_i`.)

Si l'exposant ou l'indice compte plus d'un caractère, il faut regrouper le tout entre accolades `{ }`.

`| x^{2k + 1}` x^{2k+1} `| x_{i,j}` $x_{i,j}$ `| x_{ij}^{2n}` x_{ij}^{2n}

Toutes les combinaisons d'exposants et d'indices sont possibles, y compris les puissances de puissances ou les indices d'indices.

`| e^{-x^2}` e^{-x^2} `| A_{i_s, k^n}^{y_i}` $A_{i_s, k^n}^{y_i}$



Les commandes `^` et `_` sont permises dans le mode mathématique seulement. Si \TeX rencontre l'une de ces commandes en mode texte, il tente automatiquement de passer au mode mathématique après avoir émis l'avertissement

`! Missing $ inserted.`

Il est assez rare que le résultat soit celui souhaité.

7.3.2 Fractions

Il y a plusieurs façons de représenter une fraction selon qu'elle se trouve au fil du texte, dans une équation hors paragraphe ou à l'intérieur d'une autre fraction.

Pour les fractions au fil du texte, il vaut souvent mieux utiliser simplement la barre oblique « / » pour séparer le numérateur du dénominateur, quitte à utiliser des parenthèses. Par exemple, la saisie $(n + 1)/2$ génère le texte $(n + 1)/2$.

De manière plus générale, la commande

```
\frac{⟨numérateur⟩}{⟨dénominateur⟩}
```

dispose $\langle \text{numérateur} \rangle$ au-dessus de $\langle \text{dénominateur} \rangle$, séparés par une ligne horizontale. La taille des caractères s'ajuste automatiquement selon que la fraction se trouve au fil du texte ou dans une équation hors paragraphe, ainsi que selon la position de la fraction dans l'équation.

```
% taille au fil du texte
```

```
On a $z_1 = \frac{x}{y}$ et  
$z_2 = xy$.
```

On a $z_1 = \frac{x}{y}$ et $z_2 = xy$.

```
% taille hors paragraphe
```

```
On a  
\begin{equation*}  
z_1 = \frac{x}{y}  
\end{equation*}  
et $z_2 = xy$.
```

On a

$$z_1 = \frac{x}{y}$$

et $z_2 = xy$.

```
% deux tailles combinées
```

```
Soit  
\begin{equation*}  
z = \frac{\frac{x}{2} + 1}{y}.  
\end{equation*}
```

Soit

$$z = \frac{\frac{x}{2} + 1}{y}.$$

Les commandes

```
\dfrac{⟨numérateur⟩}{⟨dénominateur⟩}  
\tfrac{⟨numérateur⟩}{⟨dénominateur⟩}
```

de **amsmath** permettent de forcer une fraction à adopter la taille d'une fraction hors paragraphe (*displayed*) dans le cas de \dfrac , ou de celle d'une fraction au fil du texte (*text*) dans le cas de \tfrac . L'exemple 7.5 permet de visualiser l'effet de la commande \dfrac .



Il est parfois visuellement plus intéressant, surtout au fil du texte, d'écrire une fraction comme $1/x$ sous la forme x^{-1} .

7.3.3 Racines

La commande

`\sqrt[⟨n⟩]{⟨radicande⟩}`

construit un symbole de radical autour de $\langle radicande \rangle$, par défaut la racine carrée. Si l'argument optionnel $\langle n \rangle$ est spécifié, c'est plutôt un symbole de racine d'ordre n qui est tracé. La longueur et la hauteur du radical s'adapte toujours à celles du radicande.

`\sqrt{2}` $\sqrt{2}$ `\sqrt{625}` $\sqrt{625}$ `\sqrt[3]{8}` $\sqrt[3]{8}$

`\sqrt[n]{x + y + z}` $\sqrt[n]{x + y + z}$

`\sqrt{\frac{x + y}{x^2 - y^2}}` $\sqrt{\frac{x + y}{x^2 - y^2}}$

7.3.4 Sommes et intégrales

Les sommes et les intégrales requièrent un symbole spécial ainsi que des limites inférieures et supérieures, le cas échéant.

Les commandes `\sum` et `\int` servent respectivement à tracer les symboles de somme Σ et d'intégrale \int . Le paquetage **amsmath** fournit également des commandes comme `\iint` et `\iiint` pour obtenir des symboles d'intégrales multiples finement disposés (\iint et \iiint).

Les éventuelles limites inférieures et supérieures sont saisies comme des indices et des exposants.

`\sum_{i = 0}^n x_i` $\sum_{i=0}^n x_i$

`\int_0^{10} f(x) \, dx` $\int_0^{10} f(x) dx$

`\iint_D f(x, y) \, dx \, dy` $\iint_D f(x, y) dx dy$

La taille des symboles et la position des limites s'ajustent automatiquement selon le contexte. Au fil du texte, la somme et l'intégrale simple ci-dessus apparaîtraient comme $\sum_{i=0}^n x_i$ et $\int_0^{10} f(x) dx$.

Dans une intégrale il est recommandé de séparer l'intégrande de l'opérateur de différentiation dx par une espace fine. C'est ce à quoi sert la commande `\,` ci-dessus; voir aussi le [tableau 7.2](#).

TAB. 7.1 – Points de suspension

commande	type de points	exemple
<code>\dots</code>	sélection automatique	
<code>\ldots</code>	points à la ligne de base	x_1, \dots, x_n
<code>\cdots</code>	points centrés	$x_1 + \cdots + x_n$
<code>\vdots</code>	points verticaux	x_1 \vdots x_n
<code>\ddots</code>	points diagonaux	x_1 \ddots x_n

7.3.5 Points de suspension

Les formules mathématiques comportent fréquemment des points de suspension dans des suites de variables ou d'opérations. Les spécialistes recommandent d'éviter de les entrer comme trois points finaux consécutifs, car l'espacement entre les points sera trop petit et le résultat, jugé disgracieux d'un point de vue typographique¹ : « ... ».

Le [tableau 7.1](#) fournit les commandes \LaTeX servant à générer divers types de points de suspension.

Avec `amsmath`, la commande `\dots` tâche de sélectionner automatiquement entre les points à la ligne de base ou les points centrés selon le contexte. Comme le résultat est en général le bon, je recommande d'utiliser principalement cette commande pour insérer des points de suspension en mode mathématique.

`x_1, \dots, x_n` x_1, \dots, x_n

`x_1, \ldots, x_n` x_1, \dots, x_n

`$x_1 + \dots + x_n$` $x_1 + \cdots + x_n$

`$x_1 + \cdots + x_n$` $x_1 + \cdots + x_n$

Le paquetage définit également les commandes sémantiques suivantes :

1. Le résultat exact dépend de la police utilisée.

- ▶ `\dotsc` pour des « points avec des virgules » (*commas*);
- ▶ `\dotspb` pour des « points avec des opérateurs binaires »;
- ▶ `\dotsm` pour des « points de multiplication »;
- ▶ `\dotsi` pour des « points avec des intégrales »;
- ▶ `\dotso` pour d'« autres points » (*other*).

7.3.6 Texte et espaces

Tel que mentionné précédemment, en mode mathématique \LaTeX traite les lettres comme des variables et gère automatiquement l'espacement entre les divers symboles. Or, il n'est pas rare que des formules mathématiques contiennent du texte (notamment des mots comme « où », « si », « quand »). De plus, il est parfois souhaitable de pouvoir ajuster les blancs entre des éléments.

La commande de `amsmath`

```
\text{<texte>}
```

insère `<texte>` dans une formule mathématique. Le texte est inséré tel quel, sans aucune gestion des espaces avant ou après le texte. Si des espaces sont nécessaires, ils doivent faire partie de `<texte>`.

```
f(x) = a e^{-ax}
\text{ pour } x > 0
 $f(x) = ae^{-ax}$  pour  $x > 0$ 
```

Les commandes

```
\quad
\qquad
```

insèrent un blanc de largeur variable selon la taille de la police en vigueur. La commande `\quad` insère un blanc de 1 em (la largeur de la lettre M dans la police en vigueur), alors que `\qquad` insère le double de cette longueur².

```
f(x) = a e^{-ax},
\quad x > 0
 $f(x) = ae^{-ax}, \quad x > 0$ 
```

Le [tableau 7.2](#) répertorie et compare les différentes commandes qui permettent d'insérer des espaces plus ou moins fines entre des éléments dans le mode mathématique.

². Bien qu'elles soient surtout utilisées dans le mode mathématique, les commandes `\quad` et `\qquad` sont également valides dans le mode texte.

TAB. 7.2 - Espaces dans le mode mathématique

Commande	Longueur	Exemple
	pas d'espace	$+ -$
<code>\,</code>	3/18 de quad	$+ -$
<code>\:</code>	4/18 de quad	$+ -$
<code>\;</code>	5/18 de quad	$+ -$
<code>\!</code>	-3/18 de quad	$+ -$
<code>\quad</code>	1 em	$+ \quad -$
<code>\qquad</code>	2 em	$+ \qquad -$

7.3.7 Fonctions et opérateurs

Les règles de typographie pour les équations mathématiques exigent que les variables apparaissent en *italique*, mais que les noms de fonctions, eux, apparaissent en romain, comme le texte standard. Pensons, ici, à des fonctions comme sin ou log.

N'oubliez pas que \LaTeX interprétera un nom de fonction saisi tel quel comme un produit de variables :

`$2 sin 30$` $2sin30$ `$3 log 2$` $3log2$

La commande `\text` permettrait d'afficher les noms de fonction en romain, mais l'utiliser à répétition se révélerait rapidement peu pratique. De plus, un nom de fonction n'est pas du texte à proprement parler, mais plutôt un opérateur mathématique.

La solution de \LaTeX consiste à définir des commandes pour un grand nombre de fonctions et d'opérateurs mathématiques standards :

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

L'espacement autour des fonctions et opérateurs est géré par \LaTeX .


`$2 \sin30$` $2 \sin 30$ `$3 \log 2$` $3 \log 2$

Certaines des fonctions ci-dessus, notamment `\lim`, acceptent des limites comme les symboles de somme et d'intégrale.

<code>% au fil du texte</code>	<code>\lim_{x \to 0} x = 0</code>	$\lim_{x \rightarrow 0} x = 0$
--------------------------------	-----------------------------------	--------------------------------

<code>% hors paragraphe</code>	<code>\lim_{x \to 0} x = 0</code>	$\lim_{x \rightarrow 0} x = 0$
--------------------------------	-----------------------------------	--------------------------------

■ amsmath

La commande `\DeclareMathOperator` de **amsmath** permet de définir de nouveaux opérateurs mathématiques lorsque nécessaire; consultez au besoin la section 5.1 de la [documentation](#)  du paquetage.

Exemple 7.1. Le matériel passé en revue jusqu'à maintenant permet déjà de composer des équations élaborées — sous réserve qu'elles tiennent sur une seule ligne comme dans le présent exemple.

Découpons pièce par pièce le code \LaTeX pour créer l'équation suivante :

$$\int_x^\infty (y - x) f_{X|X>x}(y) dy = \frac{1}{1 - F_X(x)} \int_x^\infty (y - x) f_X(y) dy.$$

<code>\begin{equation*}</code>	<i>équation hors paragraphe</i>
--------------------------------	---------------------------------

<code>\int_x^\infty</code>	\int_x^∞
----------------------------	-----------------

<code>(y - x) f_{X X > x}(y)\,</code>	$(y - x) f_{X X>x}(y) dy =$
--	-----------------------------

<code>\frac{1}{1 - F_X(x)}</code>	$\frac{1}{1 - F_X(x)}$
-----------------------------------	------------------------

<code>\int_x^\infty (y - x) f_X(y)\,</code>	$\int_x^\infty (y - x) f_X(y) dy$
---	-----------------------------------

<code>\end{equation*}</code>	<i>fin de l'environnement</i>
------------------------------	-------------------------------

□

7.4 Symboles mathématiques

Outre les chiffres et les lettres de l'alphabet, les claviers d'ordinateurs ne comptent normalement que les symboles mathématiques suivants :

+ - = < > / : ! ' | [] () { }

Vous pouvez les employer directement dans les équations, sauf les accolades $\{ \}$ qui sont des symboles réservés par \LaTeX . Il faut donc entrer celles-ci avec $\{$ et $\}$, comme dans du texte normal.

Pour représenter les innombrables autres symboles mathématiques, il faut avoir recours à des commandes qui débutent, comme d'habitude, par le symbole « \backslash » et dont le nom est habituellement dérivé de la signification mathématique du symbole.

Si un symbole mathématique a été utilisé quelque part dans une publication, il y a de fortes chances que sa version existe dans \LaTeX . Il serait donc utopique de tenter de faire ici une recension des symboles disponibles. Je me contente d'un avant-gout des principales catégories.

L'ouvrage de référence pour connaître les symboles disponibles dans \LaTeX est la bien nommée *Comprehensive \LaTeX Symbol List* (Pakin, 2021). La liste comprend plus de 18 000 symboles ! On y trouve de tout, des symboles mathématiques aux pictogrammes de météo ou d'échecs, en passant par... des figurines des Simpsons.



Les moteurs modernes $X\LaTeX$ et $\Lua\TeX$ supportent nativement le texte en format Unicode UTF-8. Ce standard contient des définitions pour plusieurs symboles mathématiques (Wikipedia, 2023). Cela signifie qu'il est possible d'entrer une partie au moins des équations mathématiques avec des caractères visibles à l'écran, plutôt qu'avec des commandes \LaTeX .

7.4.1 Lettres grecques

On obtient les lettres grecques en \LaTeX avec des commandes correspondant au nom de chaque lettre. Lorsque la commande débute par une capitale, on obtient une lettre grecque majuscule. Les commandes de certaines lettres grecques majuscules n'existent pas lorsque celles-ci sont identiques aux lettres romaines.

Les tableaux 7.3 et 7.4 présentent l'ensemble des lettres grecques disponibles dans \LaTeX .

7.4.2 Lettres modifiées

Les lettres de l'alphabet, principalement en majuscule, servent parfois en mathématiques dans des versions modifiées pour représenter des quantités, notamment les ensembles.

TAB. 7.3 - Lettres grecques minuscules

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

TAB. 7.4 - Lettres grecques 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>		

La commande `\mathcal` permet de transformer un ou plusieurs caractères en version dite « calligraphique » dans le mode mathématique.

`\mathcal{ABC}, xyz` $\mathcal{A}BC xyz$

La commande `\mathbb` fournie par les paquetages **amsfonts** et **unicode-math**, entre autres, génère des versions majuscule ajourée (*blackboard bold*) des lettres de l'alphabet. Elles sont principalement utilisée pour représenter les ensembles de nombres.

`\mathbb{NZRC}` $\mathbb{N}ZRC$



Certaines polices OpenType contiennent plusieurs versions des symboles mathématiques. Par exemple, la police utilisée dans le présent document contient deux versions de la police calligraphique, celle présentée ci-dessus et celle-ci : $\mathcal{A}\mathcal{B}\mathcal{C} xyz$. Consultez la documentation de la police pour connaître les options disponibles.

7.4.3 Opérateurs binaires et relations

Les opérateurs binaires combinent deux quantités pour en former une troisième; pensons simplement aux opérateurs d'addition $+$ et de soustrac-

TAB. 7.5 - Quelques opérateurs binaires

\times	<code>\times</code>	\div	<code>\div</code>	\pm	<code>\pm</code>	\cdot	<code>\cdot</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\setminus	<code>\setminus</code>	\circ	<code>\circ</code>
\wedge	<code>\wedge</code>	\vee	<code>\vee</code>	\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>
$*$	<code>\ast</code>	\star	<code>\star</code>	\boxplus	<code>\boxplus</code>	\boxtimes	<code>\boxtimes</code>

[†] requiert **amsmath**

TAB. 7.6 - Quelques relations et leur négation

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\neq	<code>\neq</code>	\equiv	<code>\equiv</code>
\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>	\in	<code>\in</code>	\notin	<code>\notin</code>
\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\nlessdot	<code>\nlessdot</code>	\ngeqdot	<code>\ngeqdot</code>

[†] requiert **amsmath**

tion – que l'on retrouve sur un clavier d'ordinateur normal. Les relations, quant à elles, servent pour la comparaison entre deux quantités, comme $<$ et $>$. Le [tableau 7.5](#) présente une sélection d'opérateurs binaires et le [tableau 7.6](#) une sélection de relations.

La *Comprehensive L^AT_EX Symbol List* [☞](#) consacre plus d'une dizaine de tableaux aux opérateurs binaires et près d'une quarantaine aux relations. C'est dire à quel point les tableaux 7.5 et 7.6 de la [page 93](#) ne présentent que les principaux éléments à titre indicatif.

Certaines relations existent directement en version opposée, ou négative (comme \neq ou \notin) soit dans L^AT_EX de base, soit avec **amsmath** ou un autre paquetage. Autrement, il est possible de préfixer toute relation de `\not` pour y superposer une barre oblique « / ».

7.4.4 Flèches

Les flèches de différents types sont souvent utilisées en notation mathématique, notamment dans les limites ou pour les expressions logiques. Le [tableau 7.7](#) en présente une sélection.

On retrouve les flèches utilisables en notation mathématique dans la *Comprehensive L^AT_EX Symbol List* [☞](#). Le document contient divers autres types de flèches, mais celles-ci ne sont généralement pas appropriées pour les mathématiques (pensons à \curvearrowright ou \rightarrow).

Le paquetage **amsmath** fournit plusieurs flèches additionnelles ainsi que la négation des plus communes. Ces dernières apparaissent d'ailleurs dans le [tableau 7.7](#).

TAB. 7.7 - Quelques flèches et leur négation

\leftarrow	<code>\leftarrow</code>	<code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\Leftarrow	<code>\Leftarrow</code>		\Lleftarrow	<code>\Lleftarrow</code>
\rightarrow	<code>\rightarrow</code>	<code>\to</code>	\longrightarrow	<code>\longrightarrow</code>
\Rightarrow	<code>\Rightarrow</code>		\Longrightarrow	<code>\Longrightarrow</code>
\uparrow	<code>\uparrow</code>		\downarrow	<code>\downarrow</code>
\Uparrow	<code>\Uparrow</code>		\Downarrow	<code>\Downarrow</code>
\Updownarrow	<code>\Updownarrow</code>		\Updownarrow	<code>\Updownarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>		\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>		\Leftrightarrow	<code>\Leftrightarrow</code>
\nleftarrow	<code>\nleftarrow</code>	[†]	\nleftarrow	<code>\nleftarrow</code>
\nrightarrow	<code>\nrightarrow</code>	[†]	\nrightarrow	<code>\nrightarrow</code>
\nleftrightarrow	<code>\nleftrightarrow</code>	[†]	\nleftrightarrow	<code>\nleftrightarrow</code>

[†] requiert `amsmath`

TAB. 7.8 - Accents et symboles mathématiques divers

\hat{a}	<code>\hat{a}</code>	\bar{a}	<code>\bar{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\ddot{a}	<code>\ddot{a}</code>
∞	<code>\infty</code>	∇	<code>\nabla</code>	∂	<code>\partial</code>	ℓ	<code>\ell</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\emptyset	<code>\emptyset</code>	$'$	<code>\prime</code>
\neg	<code>\neg</code>	\backslash	<code>\backslash</code>	\parallel	<code>\parallel</code>	\angle	<code>\angle</code>

7.4.5 Accents et autres symboles utiles

Le [tableau 7.8](#) présente quelques uns des accents disponibles dans le mode mathématique, ainsi que divers symboles fréquemment utilisés en mathématiques. Pour connaître l'ensemble des accents du mode mathématique de \LaTeX , les versions extensibles de certains et divers autres symboles mathématiques, consultez la section 3 de la [Comprehensive \$\text{\LaTeX}\$ Symbol List](#).

Exemple 7.2. L'équation suivante contient plusieurs des éléments présentés dans cette section et la précédente :

$$\frac{\Gamma(\alpha)}{\lambda^\alpha} = \sum_{j=0}^{\infty} \int_j^{j+1} x^{\alpha-1} e^{-\lambda x} dx, \quad \alpha > 0 \text{ et } \lambda > 0.$$

`\begin{equation*}` *équation hors paragraphe*

`\frac{\Gamma(\alpha)}{\lambda^\alpha} =`


```

\sum_{j = 0}^{\infty} \int_j^{j+1}
x^{\alpha - 1} e^{-\lambda x} dx
, \quad \alpha > 0 \text{ et }
\lambda > 0.
\end{equation*}

```

fin de l'environnement

□

7.5 Équations sur plusieurs lignes et numérotation

Dans ce qui précède, je n'ai présenté que des équations tenant sur une seule ligne en mode hors paragraphe. Cette section se penche sur la manière de représenter des groupes d'équations du type

$$y = 2x + 4 \tag{7.2}$$

$$y = 6x - 1 \tag{7.3}$$

ou des suites d'équations comme

$$\begin{aligned}
 x_{\max} &= \sum_{i=0}^{m-1} (b-1)b^i \\
 &= (b-1) \sum_{i=0}^{m-1} b^i \\
 &= b^m - 1.
 \end{aligned}$$

Je recommande fortement les environnements de **amsmath** pour les équations sur plusieurs lignes : ils sont plus polyvalents, plus simples à utiliser et leur rendu est meilleur. Le [tableau 7.9](#) — repris presque intégralement de la documentation de ce paquetage — compare les différents environnements pour les équations hors paragraphe.

- ▶ L'environnement de base pour les équations alignées sur un symbole de relation (en une ou plusieurs colonnes) est `align`. C'est l'environnement le plus utilisé en mode mathématique hormis `equation`.
- ▶ Les environnements `multline`, `gather` et `align` existent également en version étoilée (`multline*`, `gather*`, `align*`) qui ne numérotent pas les équations.

TAB. 7.9 - Comparaison des environnements pour les équations hors paragraphe de **amsmath** (les lignes verticales indiquent les marges logiques).

<pre>\begin{equation*} a = b \end{equation*}</pre>	$a = b$
<pre>\begin{equation} a = b \end{equation}</pre>	$a = b \quad (4)$
<pre>\begin{equation} \label{eq:5} \begin{split} a &= b + c - d \\ & + e - f \\ &= g + h \\ &= i \end{split} \end{equation}</pre>	$\begin{aligned} a &= b + c - d \\ &\quad + e - f \\ &= g + h \\ &= i \end{aligned} \quad (5)$
<pre>\begin{multline} a + b + c + d + e + f \\ + i + j + k + l + m + n \end{multline}</pre>	$\begin{aligned} a + b + c + d + e + f \\ + i + j + k + l + m + n \end{aligned} \quad (6)$
<pre>\begin{gather} a_1 = b_1 + c_1 \\ a_2 = b_2 + c_2 - d_2 + e_2 \end{gather}</pre>	$a_1 = b_1 + c_1 \quad (7)$ $a_2 = b_2 + c_2 - d_2 + e_2 \quad (8)$
<pre>\begin{align} a_1 &= b_1 + c_1 \\ a_2 &= b_2 + c_2 - d_2 + e_2 \end{align}</pre>	$a_1 = b_1 + c_1 \quad (9)$ $a_2 = b_2 + c_2 - d_2 + e_2 \quad (10)$
<pre>\begin{align} a_{11} &= b_{11} & \\ a_{12} &= b_{12} \\ a_{21} &= b_{21} & \\ a_{22} &= b_{22} + c_{22} \end{align}</pre>	$a_{11} = b_{11} \quad a_{12} = b_{12} \quad (11)$ $a_{21} = b_{21} \quad a_{22} = b_{22} + c_{22} \quad (12)$

- ▶ Dans les environnements `align` et `split`, les équations successives sont alignées sur le caractère se trouvant immédiatement après le marqueur de colonne « & ».
- ▶ Comme dans les tableaux, la commande `\\` sert à délimiter les lignes de la suite d'équations.
- ▶ Dans le troisième exemple du [tableau 7.9](#), la commande `\phantom` sert à insérer un blanc exactement de la largeur du symbole « = » au début de la seconde ligne de la suite d'égalités.
- ▶ Pour supprimer la numérotation d'une ligne dans une série d'équations numérotées, placer la commande `\notag` juste avant la commande de changement de ligne `\\`.

```
\begin{align}
  a_1 &= b_1 + c_1 \notag \\
  a_2 &= b_2 + c_2 - d_2 + e_2
\end{align}
```

$$a_1 = b_1 + c_1$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (7.13)$$

- ▶ Les renvois vers des équations numérotées fonctionnent, comme partout ailleurs en \LaTeX , avec le système d'étiquettes et de références ([section 6.4](#)). Le paquetage **amsmath** fournit également la très utile commande `\eqref` qui place automatiquement le numéro d'équation entre parenthèses.

```
L'équation \eqref{eq:5} du
tableau 7.9 démontre que...
```

L'équation (5) du tableau 7.9 démontre que...

- ▶ L'environnement `split` sert à apposer un seul numéro à une équation affichée sur plusieurs lignes. Il doit être employé à l'intérieur d'un autre environnement d'équations hors paragraphe.

Consultez le chapitre 3 de la [documentation](#) du paquetage **amsmath** pour les détails sur l'utilisation des environnements du tableau 7.9.

 **amsmath**

Exemple 7.3. J'ai réalisé les deux suites d'équations au début de la section avec les extraits de code ci-dessous, dans l'ordre.

```
\begin{align}
  y &= 2x + 4 \\
  y &= 6x - 1
\end{align}
```

```

\begin{align*}
x_{\text{max}}
&= \sum_{i=0}^{m-1} (b-1) b^i \\
&= (b-1) \sum_{i=0}^{m-1} b^i \\
&= b^m - 1.
\end{align*}

```

□

Veillez à respecter les règles suivantes pour la composition des équations.

1. Qu'elles apparaissent en ligne ou hors paragraphe, les équations font partie intégrante de la phrase. Ainsi, les règles de ponctuation usuelles s'appliquent aux équations.
2. Lorsqu'une équation s'étend sur plus d'une ligne, coupez chaque ligne *avant* un opérateur de sorte que chaque ligne constitue une expression mathématique complète (voir les troisième et quatrième exemples du [tableau 7.9](#)).
3. Ne numérotez que les équations d'un document auxquelles le texte fait référence.

7.6 Délimiteurs de taille variable

Les délimiteurs en mathématiques sont des symboles généralement utilisés en paire tels que les parenthèses (), les crochets [] ou les accolades { } et qui servent à regrouper des termes d'une équation. La taille des délimiteurs doit s'adapter au contenu entre ceux-ci afin d'obtenir, par exemple,

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

plutôt que la peu esthétique composition

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

La paire de commandes

```

\left<delim_g> ... \right<delim_d>

```

définit un délimiteur gauche $\langle delim_g \rangle$ et un délimiteur droit $\langle delim_d \rangle$ dont la taille s'ajustera automatiquement au contenu entre les deux commandes.

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

$$\left(\sum_{i=1}^n x_i^2 \right)^{1/2}$$

Les commandes `\left` et `\right` doivent toujours former une paire, c'est-à-dire qu'à *toute* commande `\left` doit absolument correspondre une commande `\right`. Cette contrainte est facile à oublier!

- Il est possible d'imbriquer des paires de commandes les unes à l'intérieur des autres, pour autant que l'expression compte toujours autant de `\left` que de `\right`.

$$\left[\int \left(1 + \frac{x}{k} \right) dx \right]$$

- Les symboles `\langle delim_g \rangle` et `\langle delim_d \rangle` n'ont pas à former une paire logique; toute combinaison de délimiteurs est valide.

$$\int_0^1 x \, dx = \left[\frac{x^2}{2} \right]_0^1$$

- Il arrive qu'un seul délimiteur soit nécessaire. Pour respecter la règle de la paire ci-dessus, on aura recours dans ce cas à un délimiteur *invisible* représenté par le caractère « . ».

$$f(x) = \left\{ \begin{array}{ll} 1 - x, & \text{\quad } x < 1 \\ x - 1, & \text{\quad } x \geq 1 \end{array} \right.$$

(L'environnement `aligned` utilisé ci-dessus provient de **amsmath**.) L'environnement `cases` de **amsmath** rend plus simple la réalisation de constructions comme celle ci-dessus.

```
f(x) =
\begin{cases}
1 - x, & \& x < 1 \\
x - 1, & \& x \geq 1
\end{cases}
```

$$f(x) = \begin{cases} 1 - x, & x < 1 \\ x - 1, & x \geq 1 \end{cases}$$

- La règle de la paire est tout spécialement délicate dans les équations sur plusieurs lignes car elle s'applique à *chaque ligne* d'une équation. Par conséquent, si la paire de délimiteurs s'ouvre sur une ligne et se referme sur une autre, il faudra ajouter un délimiteur invisible à la fin de la première ligne ainsi qu'au début de la seconde.

```
\begin{align*}
a
&= \left(
b + \frac{c}{d}
\right. \\
&\phantom{=} \left.
+ \frac{e}{d} - f
\right)
\end{align*}
```

$$a = \left(b + \frac{c}{d} + \frac{e}{d} - f \right)$$

- Il est également possible de choisir la taille des délimiteurs avec les commandes `\big`, `\Big`, `\bigg` et `\Bigg`. Ces commandes s'utilisent comme `\left` et `\right` en les faisant immédiatement suivre d'un délimiteur. Le [tableau 7.10](#) fournit des exemples de délimiteurs pour chaque taille.

Exemple 7.4. Le développement de la formule d'approximation de Simpson comporte plusieurs des éléments discutés jusqu'à maintenant :

$$\begin{aligned} \int_a^b f(x) dx &\approx \sum_{j=0}^{n-1} \int_{x_{2j}}^{x_{2(j+1)}} f(x) dx \\ &= \frac{h}{3} \sum_{j=0}^{n-1} [f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2(j+1)})] \end{aligned}$$

TAB. 7.10 - Tailles des délimiteurs mathématiques

taille standard	() [] { }
\big	() [] { }
\Big	() [] { }
\bigg	() [] { }
\Bigg	() [] { }

$$\begin{aligned}
 &= \frac{h}{3} \left[f(x_0) + \sum_{j=1}^{n-1} f(x_{2j}) + 4 \sum_{j=0}^{n-1} f(x_{2j+1}) \right. \\
 &\quad \left. + \sum_{j=0}^{n-2} f(x_{2(j+1)}) + f(x_{2n}) \right] \\
 &= \frac{h}{3} \left[f(a) + 2 \sum_{j=1}^{n-1} f(x_{2j}) + 4 \sum_{j=0}^{n-1} f(x_{2j+1}) + f(b) \right].
 \end{aligned}$$

J'ai composé ce bloc d'équations avec le code source ci-dessous.

```

\begin{align*}
\int_a^b f(x)\, dx
&\approx \sum_{j=0}^{n-1} \int_{x_{2j}}^{x_{2(j+1)}} f(x)\, dx \\
&= \frac{h}{3} \sum_{j=0}^{n-1} \left[ f(x_{2j}) + 4 f(x_{2j+1}) + f(x_{2(j+1)}) \right] \\
&= \frac{h}{3} \left[ f(x_0) + \sum_{j=1}^{n-1} f(x_{2j}) + 4 \sum_{j=0}^{n-1} f(x_{2j+1}) \right. \\
&\quad \left. + \sum_{j=0}^{n-2} f(x_{2(j+1)}) + f(x_{2n}) \right]
\end{align*}

```

```

\right] \\
&= \frac{h}{3}
\left[
f(a) +
2 \sum_{j = 1}^{n - 1} f(x_{2j}) +
4 \sum_{j = 0}^{n - 1} f(x_{2j + 1}) +
f(b)
\right].
\end{align*}

```

□



La section 14 de la documentation de **amsmath** traite de divers enjeux typographiques en lien avec les délimiteurs, en plus d'y introduire des nouvelles commandes pour contrôler leur taille.

De manière générale, il est déconseillé de scinder une suite d'équations entre deux pages. Le chargement du paquetage **amsmath** désactive d'ailleurs cette possibilité. Cependant, c'est parfois inévitable pour les longs blocs d'équations.

La commande `\displaybreak`, placée immédiatement avant «`\\`» dans un environnement comme `align` permet d'indiquer à \LaTeX la possibilité d'insérer un saut de page après la ligne courante dans le bloc d'équations. La commande accepte en argument optionnel un entier entre 0 et 4 indiquant à quel point un saut de page est désiré : `\displaybreak[0]` signifie « il est permis de changer de page ici » sans que ce ne soit obligatoire ; `\displaybreak[4]` ou, de manière équivalente, `\displaybreak` force un saut de page.



À moins d'en être vraiment aux toutes dernières étapes d'édition d'un document, utilisez la commande `\displaybreak` avec parcimonie et avec un argument optionnel faible.

7.7 Caractères gras en mathématiques

Les caractères gras sont parfois utilisés en mathématiques, particulièrement pour représenter les vecteurs et les matrices :

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

Pourquoi consacrer une section spécialement à cette convention typographique ? Parce que la création de symboles mathématiques en gras doit certainement figurer parmi les questions les plus fréquemment posées par les utilisateurs de \LaTeX ... et que la réponse n'est pas unique !

La commande

```
\mathbf{\langle symbole \rangle}
```

place $\langle \text{symbole} \rangle$ en caractère gras en mode mathématique. C'est donc l'analogue de la commande `\textbf` du mode texte. Dans \LaTeX de base, la commande n'a toutefois un effet que sur les lettres latines et, parfois, les lettres grecques majuscules.

```
\theta \mathbf{\theta} +
\Gamma \mathbf{\Gamma} \mathbf{+}      \theta\theta + \Gamma\Gamma + AA
A \mathbf{A}
```

On remarquera aussi que `\mathbf{A}` produit une lettre majuscule droite plutôt qu'en italique.

La manière la plus standard et robuste d'obtenir des symboles mathématiques (autres que les lettres) en gras semble être, au moment d'écrire ces lignes, via la commande


```
\bm{\langle symbole \rangle}
```

fournie par le paquetage **bm** (Carlisle, 2014).

```
\theta \bm{\theta} +
\Gamma \bm{\Gamma} \bm{+}      \theta\theta + \Gamma\Gamma + AA
A \bm{A}
```

Les utilisateurs de \XeLaTeX chargent normalement le paquetage **unicode-math** (Robertson et collab., 2020) pour sélectionner la police pour les mathématiques. Ce paquetage fournit la commande

```
\symbf{\langle symbole \rangle}
```

pour placer un $\langle \text{symbole} \rangle$ mathématique en gras. Le paquetage offre différentes combinaisons de lettres latines et grecques droites ou italiques en gras selon la valeur de l'option `bold-style`; consulter la section 5 de la [documentation](#) .

```
% XeLaTeX + paquetage unicode-math
% avec l'option bold-style=ISO
\theta \symbf{\theta} +           $\theta + \Gamma + AA$ 
\Gamma \symbf{\Gamma} \symbf{+}
A \symbf{A}
```

Si le gras est fréquemment utilisé dans un document pour une notation particulière, il est fortement recommandé de définir une nouvelle commande³ sémantique plutôt que d'utiliser à répétition l'une ou l'autre des commandes ci-dessus. Par exemple, si le gras est utilisé pour les vecteurs et matrices, vous pourriez définir une nouvelle commande `\mat` en insérant dans le préambule du document

```
\newcommand[1]{\mat}{\bm{#1}}
```

Exemple 7.5. Le paquetage `amsmath` fournit quelques environnements qui facilitent la mise en forme de matrices; ils diffèrent simplement par le type de délimiteur autour de la matrice. Ainsi, le bloc d'équations

$$\mathbf{J}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial f_1(\boldsymbol{\theta})}{\partial \theta_2} \\ \frac{\partial f_2(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial f_2(\boldsymbol{\theta})}{\partial \theta_2} \end{bmatrix} \\ = \left[\frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_j} \right]_{2 \times 2}, \quad i, j = 1, 2.$$

a été composé avec le code suivant :

```
\begin{align*}
\mat{J}(\mat{\theta})
&=
\begin{bmatrix}
\dfrac{\partial f_1(\mat{\theta})}{\partial \theta_1} & & \dfrac{\partial f_1(\mat{\theta})}{\partial \theta_2} \\
\dfrac{\partial f_2(\mat{\theta})}{\partial \theta_1} & & \dfrac{\partial f_2(\mat{\theta})}{\partial \theta_2}
\end{bmatrix} \\
&=
\end{align*}
```

3. La définition de nouvelles commandes est couvert plus en détail au [chapitre 9](#).

```
\left[
\frac{\partial f_i(\mat{\theta})}{\partial \theta_j}
\right]_{2 \times 2}, \quad i, j = 1, 2.
\end{align*}
```

La commande `\dfrac` (section 7.3.2) a servi à composer des grandes fractions à l'intérieur des matrices. J'ai également eu recours à la commande `\mat` ci-dessus. \square



Le paquetage **unicode-math** offre l'option ISO pour le traitement du gras dans les mathématiques. En fait, c'est toute la composition des mathématiques qui fait l'objet d'un standard ISO!

Plusieurs prescriptions du standard — comme les noms de variables en italique — sont déjà prises en compte par \LaTeX . Le respect de certaines autres règles est moins répandu, notamment celle qui veut que les constantes mathématiques dont la valeur ne change pas (ce sont des constantes, après tout) soient représentées en police droite. On pense, par exemple, au nombre d'Euler e , au nombre imaginaire $i = \sqrt{-1}$. Même chose pour les opérateurs mathématiques, dont le d dans les dérivées et les intégrales.

Voici un exemple d'intégrale composée avec ces règles :

$$\int_C \frac{e^z}{z} dz = 2\pi i.$$

Pour en savoir plus sur l'utilisation du standard ISO dans \LaTeX , consulter [Beccari \(1997\)](#) ou, pour un sommaire rapide, l'[entrée de blogue](#) [↗](#) de Nick Higham.

7.8 Exercices

7.1 Utiliser le gabarit de document `exercice-gabarit.tex` pour reproduire le texte suivant :

La dérivée de la fonction composée $f \circ g(x) = f[g(x)]$ est $\{f[g(x)]\}' = f'[g(x)]g'(x)$. La dérivée seconde du produit des fonctions f et g est

$$[f(x)g(x)]'' = f''(x)g(x) + 2f'(x)g'(x) + f(x)g''(x).$$

7.2 Composer l'équation suivante avec l'environnement `align*` :

$$\begin{aligned} f(x+h, y+k) &= f(x, y) + \left\{ \frac{\partial f(x, y)}{\partial x} h + \frac{\partial f(x, y)}{\partial y} k \right\} \\ &+ \frac{1}{2} \left\{ \frac{\partial^2 f(x, y)}{\partial x^2} h^2 + \frac{\partial^2 f(x, y)}{\partial x \partial y} kh + \frac{\partial^2 f(x, y)}{\partial y^2} k^2 \right\} \\ &+ \frac{1}{6} \{ \dots \} + \dots + \frac{1}{n!} \{ \dots \} + R_n. \end{aligned}$$

Aligner les deuxième et troisième lignes de l'équation sur divers caractères de la première ligne afin que l'équation ne dépasse pas les marges du document.

7.3 Composer à l'aide de l'environnement `cases` (section 7.6) la définition de la fonction $\tilde{f}(x)$:

$$\tilde{f}(x) = \begin{cases} 0, & x \leq c_0 \\ \frac{F_n(c_j) - F_n(c_{j-1})}{c_j - c_{j-1}} = \frac{n_j}{n(c_j - c_{j-1})}, & c_{j-1} < x \leq c_j \\ 0, & x > c_r. \end{cases}$$

Il est nécessaire d'imposer la taille des fractions dans la seconde branche de la définition à l'aide des fonctions de la section 7.3.2.

7.4 Le fichier `exercice-mathematiques.tex` contient un exemple complet de développement mathématique. Étudier le contenu du fichier puis compiler celui-ci tel quel avec `pdfLATEX` ou `XYLATEX`. Effectuer ensuite les modifications suivantes.

- Charger le paquetage `amsmaths` dans le préambule, puis remplacer `R^{\wedge}` par `$$\mathbb{R}^{\wedge}$$` à la ligne débutant par « Le domaine ».
- Dans l'équation du Jacobien de la transformation, remplacer successivement l'environnement `vmatrix` par `pmatrix`, `bmatrix`, `Bmatrix` et `Vmatrix`. Observer l'effet sur les délimiteurs de la matrice.
- Toujours dans la même matrice, composer successivement les deux fractions avec les commandes `\frac`, `\tfrac` et `\dfrac`. Observer le résultat.
- Réduire l'espacement de part et d'autre du symbole \Leftrightarrow dans la seconde équation hors paragraphe.
- À l'aide de la fonction Rechercher et remplacer de l'éditeur de texte, remplacer toutes les occurrences du symbole θ par λ .

8 Bibliographie et citations

La production de la bibliographie d'un ouvrage d'une certaine ampleur — qu'il s'agisse d'un article scientifique, d'un livre, d'une thèse — est une tâche d'une grande importance qui peut rapidement devenir laborieuse... lorsqu'elle n'est pas réalisée avec les outils appropriés.

L'ordinateur est bien meilleur qu'un humain pour accomplir certaines opérations propres à la production d'une bibliographie. Vous ne devriez vous préoccuper que de colliger les informations bibliographiques, puis de sélectionner les ouvrages à citer. La machine peut ensuite se charger :

- ▶ d'inclure dans la bibliographie tous les ouvrages cités dans le document et seulement ceux-ci;
- ▶ de trier les entrées de la bibliographie;
- ▶ de composer les entrées de manière uniforme;
- ▶ de recommencer ces opérations autant de fois que nécessaire pour un même document ou pour chaque nouveau document.

Avec en main une base de données bibliographique, la création de la bibliographie devient une tâche triviale qui ne prend guère plus que les quelques secondes de compilation nécessaires pour la composer.

8.1 Quel système utiliser ?

La gestion des citations et la composition d'une bibliographie sont des tâches hautement spécialisées. Comme la plupart des traitements de texte, \LaTeX les confie donc à des outils externes. Reste à savoir lequel de ces outils utiliser.

8.1.1 \BIBTeX et \natbib

Fort de plus de 25 années d'existence, \BIBTeX (Patasnik, 1988) est le système standard de traitement des bibliographies dans \LaTeX . Le système est

stable et prévisible — ce que certains considéreraient des bogues passent pour des caractéristiques — et, surtout, il existe un vaste catalogue de références bibliographiques en format `BIBTEX`. C'est généralement le seul format accepté par les revues scientifiques.

`BIBTEX` est principalement un système de tri d'entrées bibliographiques et d'interface avec la base de données. La présentation des citations et de la bibliographie relève d'un *style*. Les styles standards sont `plain`, `unsrt`, `alpha` et `abbrv`; nous y reviendrons à la [section 8.4](#).

Fonctionnant de pair — et exclusivement — avec `BIBTEX`, **natbib** (Daly, 2010) est un paquetage qui fournit des styles et des commandes pour composer des bibliographies dans le format auteur-date fréquemment utilisé dans les sciences naturelles et sociales¹. Le paquetage est également compatible avec les styles de citation standards mentionnés ci-dessus.

Parce qu'il est flexible et qu'il rend facile de produire des extensions compatibles, **natbib** est en quelque sorte devenu un standard *de facto* pour la composition des bibliographies.

Il existe plusieurs autres paquetages pour rencontrer des exigences particulières avec `BIBTEX` : bibliographies multiples, bibliographies par chapitre, etc. Mori (2009) en offre un bon survol. Consulter aussi la section *Bibliographies* de la foire aux questions [Frequently Asked Questions List for `TEX`](#).

8.1.2 Biber et biblatex

Un nouveau système de traitement des bibliographies dans `ℒTEX` est en émergence depuis quelques années. Il est formé du moteur de traitement Biber (Kline et Charette, 2023) et du paquetage **biblatex** (Lehman, 2023). Ensemble, ils visent tout à la fois à remplacer l'infrastructure bâtie autour de `BIBTEX` et à proposer des fonctionnalités additionnelles. Citons le support natif des caractères UTF-8 et de nombreux modes de citation, dont le mode auteur-titre populaire en sciences humaines.

Le duo Biber-**biblatex** bénéficie d'un développement récent en phase avec les technologies et les préoccupations actuelles. Certains enjoignent aux débutants de sauter dans ce train. Difficile, cependant, de dire si ce système saura s'établir comme le nouveau standard, surtout compte tenu de la masse de matériel disponible pour `BIBTEX`.

Le site `TEX-ℒTEX Stack Exchange` contient une [entrée](#) qui fournit un excellent sommaire des mérites et des inconvénients respectifs des deux systèmes de traitement de bibliographie.

1. C'est le style utilisé dans le présent document.

En l'absence d'un consensus clair, j'ai choisi de traiter dans ce chapitre à la fois du système le plus répandu et de celui avec lequel je suis le plus familier, soit la combinaison `BIBTEX` et `natbib`.

8.1.3 EndNote

EndNote est un logiciel commercial de gestion bibliographique très répandu dans certaines disciplines scientifiques. Il n'est donc pas rare que les nouveaux utilisateurs de `LATEX` demandent : « puis-je utiliser EndNote pour ma bibliographie ? » La réponse courte est « Non », car `LATEX` ne peut traiter directement les données bibliographiques de EndNote. La réponse plus longue est « Oui, indirectement », car EndNote possède un filtre pour exporter ses données en format `BIBTEX`.

Il est hors de la portée de ce document de traiter de la conversion de données bibliographiques de EndNote. Une simple recherche dans Internet sur « EndNote BibTeX » devrait fournir toute l'information nécessaire pour réaliser la conversion.

8.2 Processus de création d'une bibliographie

La création d'une bibliographie compte plusieurs étapes. Je les présente ici afin d'en avoir une vue d'ensemble avant d'aborder les détails dans les sections suivantes.

1. Construire une ou plusieurs bases de données contenant les informations bibliographiques. Vous souhaitez utiliser les mêmes bases de données pour tous vos documents. Par conséquent, le temps consacré à cette étape s'amenuisera au fur et à mesure que vous garnirez vos bases de données.
2. Choisir un style de citation et de présentation de la bibliographie, généralement en vous conformant aux us et coutumes de votre discipline scientifique ou aux instructions d'un éditeur.
3. Insérer dans le texte d'un document des références à des ouvrages se trouvant dans vos bases de données bibliographiques.
4. Composer les références et la bibliographie avec `BIBTEX`.

8.3 Création d'une base de données

Bien qu'il soit possible de citer des références et de construire une bibliographie avec `LATEX` sans avoir recours à une base de données bibliographiques,

je vous recommande fortement de consacrer temps et efforts à construire une telle base de données et à apprendre à utiliser `BIBTEX` pour traiter ses enregistrements. L'investissement demeure relativement faible au regard des avantages :

- ▶ les informations sont saisies une fois dans une base de données, puis utilisées à répétition;
- ▶ le traitement automatisé des informations assure une présentation uniforme de celles-ci;
- ▶ vous pouvez changer le style de présentation de la bibliographie est sans pour autant toucher aux informations bibliographiques.

La base de données n'est en fait qu'un simple fichier texte dans lequel sont regroupées dans un format précis les informations bibliographiques. Le nom du fichier doit nécessairement comporter l'extension `.bib`.

La base de données est composée d'entrées de divers *types* : livre, article scientifique, thèse, etc. Chaque entrée comporte un certain nombre de *champs* : titre, nom de l'auteur, date de publication, etc. Pour un type d'entrée donné, certains champs sont obligatoires, d'autres optionnels et d'autres simplement ignorés ou inactifs.

La structure générale d'une entrée de base de données est la suivante :

```
@⟨type_entree⟩{⟨clé⟩,
  {⟨champ⟩} = {⟨valeur⟩},
  ...,
  {⟨champ⟩} = {⟨valeur⟩}}
```

Ci-dessus, `⟨clé⟩` est un identifiant arbitraire, mais unique — et idéalement mnémonique — de l'entrée. C'est cette clé qui sera utilisée pour faire référence à l'entrée dans le code source du document.

Il est beaucoup plus facile de comprendre ce dont il est question ici par le biais d'exemples. Des commentaires et précisions additionnels sur la préparation des entrées bibliographiques suivent l'[exemple 8.1](#).

Exemple 8.1. On trouvera ci-dessous les entrées bibliographiques d'un livre (Kopka et Daly, 2003), d'un article scientifique (Mori, 2009) et d'un manuel générique, en l'occurrence la documentation d'un paquetage `LATEX` (Daly, 2010). Pour faciliter la comparaison, chaque entrée est immédiatement suivie du texte de la notice tel qu'il apparaît dans la bibliographie du présent ouvrage.

```
@book{Kopka:latex:4e,
  author = {Kopka, Helmut and Daly, Patrick W.},
  title = {Guide to {\LaTeX}},
```



```
publisher = {Addison-Wesley},  
year =      2003,  
edition =   4,  
isbn =      {978-032117385-0},  
language =  {english}}
```

Kopka, H. et P. W. Daly. 2003, *Guide to L^AT_EX*, 4^e éd., Addison-Wesley, ISBN 978-032117385-0.

```
@article{Mori:bibliographies:2009,  
  author =   {Lapo F. Mori},  
  title =    {Managing bibliographies with {\LaTeX}},  
  journal =  {{TUG}boat},  
  year =     2009,  
  volume =  30,  
  number =  1,  
  pages =   {36-48},  
  url =     {https://www.tug.org/TUGboat/tb30-1/tb94mori.pdf},  
  language = {english}}
```

Mori, L. F. 2009, « Managing bibliographies with L^AT_EX », *TUGboat*, vol. 30, n° 1, p. 36-48, <https://www.tug.org/TUGboat/tb30-1/tb94mori.pdf>.

```
@manual{natbib,  
  author =   {Patrick W. Daly},  
  title =    {Natural Sciences Citations and References},  
  year =     2010,  
  url =      {https://www.ctan.org/pkg/natbib},  
  language = {english}}
```

Daly, P. W. 2010, *Natural Sciences Citations and References*, <https://www.ctan.org/pkg/natbib>.

□

■ btxdoc

- ▶ Les types d'entrée bibliographique dans l'exemple ci-dessus sont `book`, `article` et `Manual`². Vous remarquerez que les champs utilisés sont différents d'un type à un autre. Vous trouverez la liste de tous les types d'entrée et des champs obligatoires et optionnels pour chacun dans, entre autres, la [documentation de BibTeX](#), [Wikipedia](#) ou la plupart des bons ouvrages de référence (dont [Kopka et Daly, 2003](#)).
- ▶ Le nom d'un auteur est saisi soit sous la forme `{Prénom Nom}`, soit sous la forme `{Nom, Prénom}`. La seconde forme est surtout utile pour distinguer explicitement le nom du prénom, par exemple dans le cas de prénoms ou de noms multiples.
- ▶ Lorsqu'un ouvrage compte plusieurs auteurs, on distingue ceux-ci en séparant le nom complet de *chacun* des auteurs par.
- ▶ BibTeX gère automatiquement les hauts et bas de casse (majuscules et minuscules), en particulier dans les titres d'ouvrages. Pour préserver une casse particulière, il suffit de placer les lettres entre accolades. Par exemple, dans la seconde entrée de l'[exemple 8.1](#), le titre du journal TUGboat est inscrit sous la forme `{{TUG}boat}` pour éviter que BibTeX ne le transforme en « Tugboat ».
- ▶ Les champs `isbn`, `url`, `doi` et quelques autres ([Daly, 2010](#), section 2.8) sont fournis par le paquetage **natbib**. Même si ces champs ne devaient pas s'afficher dans la bibliographie pour le style choisi, c'est une bonne idée d'insérer les informations dans la base de données pour référence future.
- ▶ Le champ `language`, introduit par **babel**, permet de préciser la langue de l'entrée bibliographique. La coupure de mots et la composition de certains éléments seront ainsi adaptées en conséquence. Par exemple, si l'entrée d'un document comporte le champ `edition = 2`, sa fiche bibliographique contiendra la mention « 2^e édition » ou « 2nd edition » selon que l'on a précisé que l'ouvrage est en français ou en anglais.
- ▶ BibTeX supporte les lettres accentuées ou autres caractères spéciaux dans le texte des champs seulement si le document utilisant la bibliographie supporte lui-même ces caractères ; voir la [section 2.10](#). Autrement, il faut entrer les caractères spéciaux avec les commandes \LaTeX de base tel qu'expliqué sommairement à la [section 2.9](#). Ma recommandation : évitez les lettres accentuées dans les entrées susceptibles d'être utilisées dans un document entièrement en anglais.

². Les identifiants des types d'entrée et des champs sont insensibles à la casse. Par exemple, on pourrait tout aussi bien débiter une entrée par `@Manual`, `@manual` ou `@MANUAL`.



Entretenir une base de données bibliographiques unique peut rapidement devenir pénible quand son nombre d'entrées devient grand. Mieux vaut alors scinder ses références dans plusieurs fichiers par thématique, un peu comme dans une bibliothèque : droit, finance, informatique, mathématiques, etc. Vous pouvez alors nommer les fichiers selon la thématique : `droit.bib`, `finance.bib`, `informatique.bib`, etc.

8.4 Style des citations et de la bibliographie

Il existe plusieurs manières différentes de présenter une bibliographie et \LaTeX sait s'adapter. Le format général de la bibliographie est contrôlé par un *style* choisi avec la commande `\bibliographystyle`. Le style affecte habituellement deux composantes de la bibliographie :

1. le mode de citation dans le texte (numérique, alphanumérique, auteur-date, etc.);
2. la présentation des notices bibliographiques (ordre des éléments, ponctuation, mise en forme des caractères, etc.).

Le [tableau 8.1](#) contient des exemples de quelques styles bibliographiques.

Les styles standards de \LaTeX sont `plain`, `unsrt`, `alpha` et `abbrv`. Ces styles ont été développés pour des modes de citation numériques ou alphanumériques.

Pour plus de flexibilité, je recommande d'utiliser le paquetage **natbib** pour la gestion des références et du style de la bibliographie. Entre autres choses, ce paquetage supporte le style de citation auteur-date fréquemment employé en sciences naturelles et sociales, plusieurs commandes de citation, un grand nombre de styles de bibliographie, ainsi que des entrées spécifiques pour les numéros ISBN, les adresses URL et les identificateurs d'objets numériques (DOI). Le paquetage fournit des styles `plainnat`, `unsrnat` et `abbrvnat` similaires aux styles standards, mais plus complets. Il existe des [versions francisées](#) de ces styles dans CTAN et dans \TeX Live. La consultation de la [documentation](#) de **natbib** est fortement recommandé. Vous y trouverez également des informations sur l'utilisation de styles de citation autres que auteur-date.

 **natbib**

Également dans CTAN et dans \TeX Live, mon paquetage **francais-bst** ([Goulet, 2023](#)) fournit deux styles bibliographiques compatibles avec **natbib** permettant de composer des bibliographies auteur-date respectant les normes de typographie française proposées dans [Malo \(1996\)](#); voir le dernier exemple du [tableau 8.1](#).

TAB. 8.1 – Quelques styles bibliographiques et leur effet sur le mode de citation et le format des notices bibliographiques

Styles standards numériques et alphanumériques

style	mode de citation	format de notice
plain	Un bon ouvrage de référence est [1]	[1] Helmut Kopka and Patrick W. Daly. <i>Guide to L^AT_EX</i> . Addison-Wesley, 4 édition, 2003.
plain-fr	Un bon ouvrage de référence est [1]	[1] Helmut KOPKA et Patrick W. DALY : <i>Guide to L^AT_EX</i> . Addison-Wesley, 4 édition, 2003.
alpha-fr	Un bon ouvrage de référence est [KDo3]	[KDo3] Helmut KOPKA et Patrick W. DALY : <i>Guide to L^AT_EX</i> . Addison-Wesley, 4 édition, 2003.

Styles auteur-date avec natbib

style	mode de citation	format de notice
plainnat-fr	Un bon ouvrage de référence est Kopka et Daly (2003)	Helmut KOPKA et Patrick W. DALY : <i>Guide to L^AT_EX</i> . Addison-Wesley, 4 édition, 2003.
français	Un bon ouvrage de référence est Kopka et Daly (2003)	Kopka, H. et P. W. Daly. 2003. <i>Guide to L^AT_EX</i> , 4 ^e éd., Addison-Wesley.

8.5 Insertion de références dans le texte

La raison première d'une bibliographie, c'est évidemment d'y colliger les informations relatives aux ouvrages auxquels un document fait référence. Avant de penser créer une bibliographie, il faut donc savoir comment insérer des références dans le texte.

La commande de base pour insérer une référence au fil du texte dans L^AT_EX est

```
\cite{<clé>}
```

L'effet de la commande est double :

1. insérer une référence comme « Mori (2009) » dans le texte ;
2. ajouter le document dans la bibliographie.

En somme, outre la phase de compilation qui fait l'objet de la section suivante, c'est tout ce qu'il y a à faire pour construire la bibliographie d'un document.

Le paquetage **natbib** propose plutôt les commandes

```
\citet{<clé>}
\citep{<clé>}
```

Dans le style de citation auteur-date, ces commandes permettent, respectivement, d'insérer une référence au fil de la phrase ou en aparté :


```
\citet{Mori:bibliographies:2009} Mori (2009) en offre un bon sur-
en offre un bon survol. vol.
```


```
TUGboat a publié un bon survol TUGboat a publié un bon survol
\citep{Mori:bibliographies:2009}. (Mori, 2009).
```

Vous ne devriez *jamais* entrer directement dans le texte des informations bibliographiques, même partielles. Pour insérer dans le texte le nom d'un auteur ou l'année de publication d'un ouvrage, utilisez les commandes de **natbib** suivantes :

```
\citeauthor{<clé>}
\citeyear{<clé>}
```

Avec ces commandes, pas de risque de mal orthographier un nom par inadvertance, ou d'oublier de modifier dans le texte une année de publication que l'on aura changé dans la base de données bibliographique.

Si vous utilisez le paquetage **natbib**, je vous recommande de jeter un œil à sa [documentation](#) . Vous y trouverez des informations sur plusieurs autres commandes pour manipuler les informations bibliographiques et contrôler leur présentation.

 natbib

Il arrive que l'on souhaite inclure dans la bibliographie un ou plusieurs documents qui ne sont pas cités dans le texte. Pour ce faire, il suffit d'insérer dans le corps du document la commande

```
\nocite{<clé1, clé2, ...>}
```

où *<clé1>*, *<clé2>*, ..., sont les clés des documents à inclure dans la bibliographie.

Le paquetage **hyperref** fait automatiquement d'une référence bibliographique un hyperlien vers l'entrée dans la bibliographie. C'est le cas dans le

présent document. Or, il peut arriver que l'hyperlien soit superflu ou indésirable. Pour le supprimer pour une référence particulière, utilisez l'environnement NoHyper :

```
\begin{NoHyper} \citet{<clé>} \end{NoHyper}
```

8.6 Composition de la bibliographie

Les commandes de la section précédente servent à indiquer à \LaTeX les ouvrages à inclure dans la bibliographie. C'est toutefois l'outil externe \BIBTeX qui se chargera de fournir à \LaTeX le texte des références ainsi que le contenu de la bibliographie.

À la [section 1.3](#), j'ai représenté schématiquement comme suit le processus de création d'un document avec \pdf\LaTeX ou \Xe\LaTeX :



Pour créer ou mettre à jour la bibliographie, il s'ajoute au processus une étape de compilation du document avec \BIBTeX :



Plus en détails, le processus de préparation d'un document comprenant une bibliographie est le suivant.

1. Composer le texte et y insérer des références avec les commandes de la section précédente.
2. Définir le style bibliographique dans le préambule ou près de la commande `\bibliography` avec la commande

```
\bibliographystyle{<style>}
```

3. Insérer dans le texte la commande

```
\bibliography{<base_donnees1>, <base_donnees2>, ...}
```

à l'endroit où doit apparaître la bibliographie (généralement à la fin du document). Les arguments $\langle base_donnees1 \rangle$, $\langle base_donnees2 \rangle$, séparés par des virgules, sont les noms (sans l'extension `.bib`) des fichiers de données bibliographiques.

4. Compiler le document une première fois afin que \LaTeX détecte les ouvrages à insérer dans la bibliographie. À cette étape, les références dans le texte apparaissent sous forme d'un point d'interrogation « ? ».
5. Compiler le document avec $\text{BIB}\TeX$ afin de préparer le texte des références et composer la bibliographie.
6. Compiler à nouveau le document au moins deux fois afin d'y insérer d'abord la bibliographie, puis le texte des références.

Il faut répéter les étapes 4-6 chaque fois qu'une nouvelle référence est ajoutée dans le document ou que l'entrée bibliographique est modifiée. Autrement, tant que la bibliographie demeure inchangée, une compilation standard suffit.

Exemple 8.2. La [figure 8.1](#) présente le contenu d'un fichier de base de données bibliographiques et le fichier source d'un document simple contenant des références et une bibliographie.

- ▶ Le document utilise le mode de citation auteur-date de **natbib**.
- ▶ La présence du paquetage **fontspec** dans le préambule indique que le document doit être compilé avec $\text{Xe}\LaTeX$. Cela fait en sorte que les lettres accentuées sont automatiquement supportées tant dans le texte que dans les entrées de la bibliographie.
- ▶ Le paquetage **babel** est activé avec l'anglais et le français, les deux langues utilisées dans la bibliographie. Nommé en dernier dans les options de chargement de la classe, le français est la langue par défaut du document.

Ces fichiers en main, les étapes de composition 1-3 sont complétées. La [figure 8.2](#) présente la zone de texte principale du document après l'étape 4, puis après chacune des deux compilations de l'étape 6. □



Lorsque vous en êtes aux toutes dernières étapes avant de publier un document, assurez-vous d'exécuter $\text{BIB}\TeX$ et de compiler le document au moins deux fois et que le journal de la compilation (*log file*) ne rapporte pas de références manquantes (*undefined references*).

exemple-bibliographie.bib

```
@Book{Kopka:latex:4e,
  author = {Kopka, Helmut and Daly, Patrick W.},
  title = {Guide to {\LaTeX}},
  publisher = {Addison-Wesley},
  year = 2003,
  edition = 4,
  isbn = {978-0321173850},
  language = {english}}

@Book{Malo:1996,
  author = {Malo, M.},
  title = {Guide de la communication écrite au cégep,
    à l'université et en entreprise},
  publisher = {Québec Amérique},
  year = 1996,
  language = {french}}
```

exemple-bibliographie.tex

```
\documentclass[11pt,english,french]{article}
  \usepackage{fontspec}
  \usepackage{natbib}
  \usepackage{babel}

  \bibliographystyle{francais}

\begin{document}

Nous recommandons \citet{Kopka:latex:4e} comme ouvrage de
référence sur {\LaTeX}.

La bibliographie est mise en forme selon des normes de
typographie française \citep{Malo:1996}.

\bibliography{exemple-bibliographie}

\end{document}
```

FIG. 8.1 - Code source d'un fichier de base de données (haut) et d'un document simple (bas)

xelatex

Nous recommandons ? comme ouvrage de référence sur \LaTeX .
 La bibliographie est mise en forme selon des normes de typographie française (?).

xelatex → bibtex → xelatex

Nous recommandons ? comme ouvrage de référence sur \LaTeX .
 La bibliographie est mise en forme selon des normes de typographie française (?).

Références

Kopka, H. et P. W. Daly. 2003, *Guide to \LaTeX* , 4^e éd., Addison-Wesley, ISBN 978-0321173850.

Malo, M. 1996, *Guide de la communication écrite au cégep, à l'université et en entreprise*, Québec Amérique.

xelatex → bibtex → xelatex → xelatex

Nous recommandons Kopka et Daly (2003) comme ouvrage de référence sur \LaTeX .

La bibliographie est mise en forme selon des normes de typographie française (Malo, 1996).

Références

Kopka, H. et P. W. Daly. 2003, *Guide to \LaTeX* , 4^e éd., Addison-Wesley, ISBN 978-0321173850.

Malo, M. 1996, *Guide de la communication écrite au cégep, à l'université et en entreprise*, Québec Amérique.

FIG. 8.2 - Zone de texte du document aux diverses étapes de la compilation des fichiers de la [figure 8.1](#) avec \XeTeX et \BibTeX

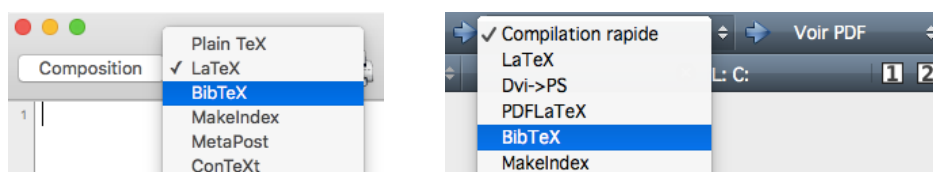


FIG. 8.3 – Interfaces de sélection du programme BibTeX dans TeXShop (à gauche) et Texmaker (à droite)

Les logiciels intégrés de rédaction offrent généralement des raccourcis pour exécuter la compilation avec BibTeX .

- ▶ Dans TeXShop, sélectionner un autre programme dans le menu à côté du bouton « Composition ».
- ▶ Dans Texmaker, choisir le programme approprié dans le menu de composition rapide.
- ▶ Dans GNU Emacs, choisir BibTeX dans le menu Command ou après avoir lancé la commande `TeX-command-master` avec `C-c C-c`.

La [figure 8.3](#) présente les deux premières interfaces.

[Wikilivres](#) consacre toute une [section](#) à la gestion de la bibliographie. Vous y trouverez plusieurs informations additionnelles, notamment sur des sources de données bibliographiques et des outils de gestion des bases de données.

8.7 Exercices

8.1 Composer des entrées de base de données pour les références bibliographiques suivantes.

- a) Mittelbach, F. 2014b, « How to influence the position of float environments like figure and table in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$? », *TUGboat*, vol. 35, n° 3, p. 258-254

Astuce : cette entrée est un article tiré d'une revue scientifique.

- b) Wilson, P. 2022, *The Memoir Class for Configurable Typesetting*, 8^e éd., The Herries Press, <https://www.ctan.org/pkg/memoir>. Maintained by Lars Madsen

Astuces : traiter cette entrée comme un livre et utiliser le champ `note` pour consigner la remarque qui se trouve à la fin de la notice.

- c) Van Zandt, T., D. Girou et H. Voß. 2014, *PSTricks — PostScript Macros for Generic T_EX*, <https://www.ctan.org/pkg/pstricks-base>
Astuces : utiliser le type de document `Manual` ; attention à la casse de certains mots ; on obtient le symbole ß avec la commande `\ss`.
- 8.2** Utiliser pour cet exercice le fichier `exercice-gabarit.tex` ainsi que la base de données bibliographique créée à l'exercice précédent.
- a) Créer un document simple comprenant des références à une ou plusieurs des entrées bibliographiques de l'exercice précédent. Compiler le document en suivant les étapes mentionnées à la [section 8.6](#) en utilisant tour à tour les styles par défaut `plain`, `unsrt`, `alpha` et `abbrv`.
- b) Charger dans le document le paquetage **natbib** (avant **babel**) et utiliser le style de bibliographie français fourni par **francais-bst** (installé par défaut dans T_EX Live). Recompiler le document et observer les différences par rapport aux documents produit en a).
- 8.3** À partir d'un gabarit fourni avec la classe `ulthese`, produire un document simple contenant une bibliographie.



9 Commandes et environnements définis par l'utilisateur

\LaTeX est un ensemble de macro-commandes conçu pour faciliter l'utilisation du système $T_{\text{E}}X$. Dès lors, les utilisateurs peuvent à leur tour définir des commandes. Il suffit généralement d'avoir rédigé quelques documents — ou quelques chapitres d'un long document — avec \LaTeX pour réaliser combien cette possibilité est de nature à faciliter le travail.

La définition de nouvelles commandes et de nouveaux environnements peut servir à créer des extensions à \LaTeX — c'est d'ailleurs ce que font plusieurs paquetages. Cependant, en usage courant, on fera principalement appel à ces fonctionnalités pour l'une ou l'autre des trois raisons suivantes :

1. créer des raccourcis pour de longues commandes utilisées fréquemment;
2. créer des commandes sémantiques afin d'uniformiser la présentation du texte;
3. modifier le comportement de commandes existantes — car il est également possible de redéfinir une commande existante.

Exemple 9.1. J'ai créé ou modifié des commandes pour chacune des raisons ci-dessus dans la préparation du présent ouvrage.

1. Une nouvelle commande `\doc` facilite et systématise l'insertion de liens vers la documentation. D'un seul appel, elle crée un hyperlien dans le texte suivi de l'icône  et elle ajoute le nom du fichier de documentation dans la marge précédé de l'icône .
2. Une nouvelle commande sémantique `\pkg` sert pour la mise en forme des noms de paquetages. Ainsi, leur présentation est toujours la même et, si je devais souhaiter en changer, il me suffirait de modifier la définition de la commande.
3. La redéfinition de la commande `\chaptitelfont` de la classe `memoir` permet de modifier la police et la mise en forme des titres de chapitres.

Je reviendrai sur les détails de ces exemples dans la suite du chapitre. \square

9.1 Nouvelles commandes

Les commandes `\newcommand` et `\renewcommand` permettent respectivement de définir une nouvelle commande et de redéfinir une commande existante — c'est-à-dire d'en modifier la définition. On place généralement ces définitions dans le préambule du document.

9.1.1 Commandes sans arguments

Certaines commandes ne requièrent pas d'argument; pensons à `\LaTeX` ou `\bfseries`. Ce sont les commandes les plus simples à créer. La syntaxe des commandes `\newcommand` et `\renewcommand` pour de tels cas est la suivante :

```
\newcommand{\langle nom_commande \rangle}{\langle définition \rangle}
\renewcommand{\langle nom_commande \rangle}{\langle définition \rangle}
```

Le premier argument, `\langle nom_commande \rangle`, est le nom de la commande, avec le symbole « `\` ». Pour `\newcommand`, ce nom doit être différent de celui de toute commande active dans le document¹. À l'inverse, avec `\renewcommand`, le nom doit déjà être pris par une commande active.

Le second argument, `\langle définition \rangle`, contient la définition de la commande. Il peut s'agir de caractères à insérer dans le texte, de commandes à exécuter ou d'une combinaison de tout cela.

Exemple 9.2. La commande `\mathbb`, présentée à la [section 7.4.2](#), permet de créer une lettre majuscule ajourée pour représenter un ensemble de nombres en mathématiques. Plutôt que de l'utiliser à divers endroits dans un document, il est préférable de définir une commande sémantique comme `\R` pour représenter l'ensemble des nombres réels :

```
\newcommand{\R}{\mathbb{R}}
```

Comme cela, si vous souhaitez pour une raison quelconque modifier la représentation de l'ensemble des nombres réels, il vous suffit de modifier la définition de la commande `\R` pour que le changement prenne effet dans tout le document. \square

¹. Les commandes actives dans un document sont les commandes de base de \TeX et \LaTeX ainsi que les commandes de tous les paquetages chargés dans le préambule.

Exemple 9.3. Tel que mentionné à l'[exemple 9.1](#), j'ai modifié la police des titres de chapitres dans le présent document en redéfinissant la commande `\chaptitfont` de la classe `memoir`. Pour obtenir des titres de chapitres sans empattements, en caractères gras, de dimension `\Huge` et alignés à gauche, le préambule du document contient la déclaration

```
\renewcommand{\chaptitfont}{\normalfont%
\sffamily\bfseries\Huge\raggedright}
```

□

9.1.2 Commandes avec arguments

Les commandes `\newcommand` et `\renewcommand` ont d'autres tours dans leur sac. Leur syntaxe étendue permet également de définir ou de redéfinir des commandes acceptant un ou plusieurs arguments :

```
\newcommand{\<nom_commande>}[<narg>]{<définition>}
\renewcommand{\<nom_commande>}[<narg>]{<définition>}
```

Le nouvel argument `<narg>` est un nombre entre 1 et 9 spécifiant le nombre d'arguments de la commande. La `<définition>` de la commande doit alors contenir des jetons `#1`, `#2`, ... pour identifier les endroits où les arguments 1, 2, ... doivent apparaître.

Exemple 9.4. La nouvelle commande `\pkg` mentionnée à l'[exemple 9.1](#) affiche les noms de paquetages en caractères gras. La commande prend en argument le nom du paquetage. Sa définition est donc

```
\newcommand{\pkg}[1]{\textbf{#1}}
```

Il s'agit encore d'une commande sémantique permettant de changer aisément la mise en forme en modifiant une seule définition dans le préambule du document. □

Exemple 9.5. La commande `\doc` mentionnée à l'[exemple 9.1](#) requiert trois arguments :

1. le texte de l'hyperlien qui sera placé au fil du texte ;
2. le nom du fichier de documentation à placer dans la marge dans une police non proportionnelle ;
3. l'adresse URL vers le fichier de documentation en ligne.

Voici une version simplifiée de la définition de la commande :

```
\newcommand{\doc}[3]{%
  \href{#3}{#1~\faExternalLink*}%
  \marginpar{\faBookmark~\texttt{#2}}}
```

La commande `\href` qui permet d'insérer un hyperlien dans le texte provient du paquetage **hyperref** (Rahtz et Oberdiek, 2023). Les commandes `\faExternalLink*` et `\faBookmark` proviennent du paquetage **fontawesome5** (Krüger, 2022); elles insèrent dans le texte des icônes de la police libre **Font Awesome**.

Avec la définition ci-dessus, la déclaration

```
\doc{documentation}{hyperref}{%
  https://texdoc.net/pkg/hyperref}
```

■ hyperref

produit : [documentation](https://texdoc.net/pkg/hyperref) 

□

9.2 Nouveaux environnements

Tel que mentionné en introduction du chapitre, \LaTeX permet également à l'utilisateur de définir ou de modifier des environnements. La mécanique est similaire à celle de la définition de commandes, sauf qu'un environnement compte trois parties : le début, marqué par la déclaration `\begin{...}` et, parfois, des commandes de configuration de l'environnement ; le contenu en tant que tel ; la fin, marquée par la déclaration `\end{...}`.

Les commandes suivantes servent à créer ou à modifier des environnements :

```
\newenvironment{<nom_env>}[<narg>]{<début_déf>}{<fin_déf>}
\renewenvironment{<nom_env>}[<narg>]{<début_déf>}{<fin_déf>}
```

Les nombreux arguments sont les suivants :

- `<nom_env>` nom de l'environnement à créer ou à modifier. Il est fortement recommandé de ne pas modifier les environnements de base de \LaTeX ;
- `<narg>` un nombre entre 1 et 9 représentant le nombre d'arguments de l'environnement, lorsqu'il y en a. Les arguments sont utilisés de la même manière que dans les définitions de commandes ;
- `<début_déf>` commandes et texte à insérer au début de l'environnement, lors de l'appel `\begin{<nom_env>}`. C'est dans ce bloc que doivent se trouver les jetons `#1`, ..., `#<narg>` lorsque l'environnement a des arguments.

$\langle fin_déf \rangle$ commandes et texte à insérer à la fin de l'environnement, lors de l'appel $\end{\langle nom_env \rangle}$.

Exemple 9.6. Créons un environnement pour composer les citations hors paragraphe de la manière suivante :

Texte en italique, dans une police de taille inférieure au texte normal et en retrait des marges gauche et droite.

Ceci est simple à réaliser en se basant sur l'environnement standard `quote` et en modifiant les attributs de police :

```
\begin{quote}
  \small\itshape%
  Texte en italique...
\end{quote}
```

Pour automatiquement composer toutes les citations de la même manière, il suffit de créer un nouvel environnement, disons `citation` :

```
\newenvironment{citation}%
  {\begin{quote}\small\itshape}%
  {\end{quote}}
```

Le bloc de code ci-dessus peut ensuite être remplacé par

```
\begin{citation}
  Texte en italique...
\end{citation}
```

□



N'hésitez pas à créer des nouvelles commandes et des nouveaux environnements dès lors qu'une mise en forme particulière revient plus d'une ou deux fois dans un document.

9.3 Exercices

9.1 Certains auteurs composent les sigles et les acronymes² en petites capitales, avec ou sans les points : C.Q.F.D., NASA.

- a) Créer une commande \backslash NASA qui insère l'acronyme NASA dans le texte. Rappelons que l'on compose du texte en petites capitales avec la commande \backslash textsc.

2. Un sigle est une abréviation formée par une suite de lettres qui sont les initiales d'un groupe de mots. Un acronyme est un sigle qui se prononce comme un mot ordinaire.

- b) Créer une commande plus générale `\sigle` qui affiche son argument en petites capitales. La commande devra convertir l'argument en minuscules avec `\MakeLowercase` afin que le résultat soit toujours le même peu importe la casse utilisée dans le code. Ainsi, `\sigle{nasa}`, `\sigle{Nasa}` et `\sigle{NASA}` donneront toujours NASA.
- c) Après avoir utilisé la commande `\sigle` à quelques reprises dans un document, modifier sa définition pour plutôt composer les sigles en majuscules.

Utiliser le gabarit de document `exercice-gabarit.tex` pour créer et tester les commandes ci-dessus.

- 9.2** Je n'ai pas abordé dans le chapitre une fonctionnalité plus avancée de `\newcommand` et `\renewcommand`, soit celle de pouvoir définir des commandes dont un argument est optionnel ou, plus précisément, de donner une valeur par défaut à un argument.

La syntaxe réellement complète de `\newcommand` et `\renewcommand` est la suivante :

```
\newcommand{\langle nom_commande \rangle}[\langle narg \rangle][\langle option \rangle]{\langle déf \rangle}
\renewcommand{\langle nom_commande \rangle}[\langle narg \rangle][\langle option \rangle]{\langle déf \rangle}
```

L'argument additionnel `\langle option \rangle` contient la valeur par défaut du *premier* argument de `\langle nom_commande \rangle`. Dès lors, la commande ne compte plus `\langle narg \rangle` arguments obligatoires, mais bien `\langle narg \rangle - 1` arguments obligatoires et un optionnel.

Modifier la définition de la commande `\doc` de l'[exemple 9.5](#) pour que « documentation » soit le texte par défaut de l'hyperlien qui est placé au fil du texte.

- 9.3** Modifier l'environnement `citation` de l'[exemple 9.6](#) afin de composer les citations hors paragraphe comme suit :

La citation est toujours en retrait des marges gauche et droite, mais également surmontée et suivie de filets horizontaux. Le texte est en police de taille `\small`, droite et sans empattements.

Astuce : utiliser un tableau pleine largeur à l'intérieur de l'environnement `quote` pour disposer le texte et créer les filets horizontaux.

10 Trucs et astuces divers

En clôture de l'ouvrage, ce chapitre traite de différents sujets que même une personne débutant avec \LaTeX voudra assez rapidement aborder, comme le contrôle des sauts de ligne et des sauts de page, la modification de la police du document, l'utilisation de la couleur ou l'insertion d'hyperliens dans le fichier de sortie PDF. J'offre également de courtes introductions à des usages plus spécialisés de \LaTeX comme la mise en page de code informatique, la production de diapositives ou la programmation lettrée. Enfin, j'explique sommairement comment assurer de manière efficace la gestion des versions de ses documents, surtout dans un contexte de travail collaboratif.

10.1 Contrôle de la disposition du texte

Des algorithmes élaborés permettent généralement à \LaTeX de disposer le texte harmonieusement sur la page. Néanmoins, il est parfois nécessaire d'effectuer soi-même de menus ajustements, notamment pour les sauts de page et la coupure de mots.

10.1.1 Sauts de ligne et de page

Il est rarement nécessaire, avec \LaTeX , de forcer les retours à la ligne. Chose certaine, vous devriez toujours utiliser une ligne blanche dans le code source pour identifier un changement de paragraphe.

Cela dit, les commandes suivantes permettent d'insérer un saut de ligne manuellement lorsque requis :

```
\\  
\\[<longueur>  
\newline
```

La commande `\\` est connue : elle sert aussi à délimiter les lignes dans les tableaux ([section 6.2](#)) et les lignes d'une suite d'équations ([section 7.5](#)). L'ar-

gument optionnel [*longueur*] permet d'insérer un blanc entre les deux lignes ; la [section 2.7](#) explique comment spécifier une longueur.

Généralement équivalente à `\\`, la commande `\newline` est parfois nécessaire, notamment pour insérer un changement de ligne à l'intérieur d'une cellule d'un tableau ou à l'intérieur d'un titre de section. Quand `\\` ne fonctionne pas, essayez `\newline`.

Exemple 10.1. La commande `\\` est particulièrement utile — voire nécessaire — pour disposer des boîtes à l'intérieur d'une figure. L'utilisation de l'argument *longueur* permet alors de contrôler l'espacement vertical entre les éléments. Comparez les deux exemples ci-dessous.

```
\begin{minipage}{1.0\linewidth}
  \framebox[\linewidth]{texte}
\end{minipage} \\
\begin{minipage}{1.0\linewidth}
  \framebox[\linewidth]{texte}
\end{minipage}
```

texte
texte

```
\begin{minipage}{1.0\linewidth}
  \framebox[\linewidth]{texte}
\end{minipage} \\[6pt]
\begin{minipage}{1.0\linewidth}
  \framebox[\linewidth]{texte}
\end{minipage}
```

texte
texte

□

Les commandes

```
\newpage
\clearpage
\cleartorecto           % classe memoir seulement
\cleartoverso          % classe memoir seulement
```

permettent d'insérer manuellement un saut de page pour éviter une coupure malheureuse. La commande de base pour insérer un saut n'importe où dans la page est `\newpage`. La commande `\clearpage`, quant à elle, va également s'assurer d'afficher tous les éléments flottants ([section 6.4](#)) en attente de disposition.

Les commandes `\cleartorecto` et `\cleartoverso`, propres à la classe `memoir` permettent respectivement de passer automatiquement à une page

recto ou à une page verso. Évidemment, elles n'ont d'utilité que dans les documents recto verso.

Moins directives, les commandes

```
\pagebreak[⟨n⟩]
\enlargethispage{⟨longueur⟩}
```

permettent de seulement aider \TeX à gérer les sauts de page à un endroit précis. La commande `\pagebreak` est intéressante lorsque utilisée avec son argument optionnel $\langle n \rangle$: celui-ci indique, par le biais d'un nombre entier entre 0 et 4, à quel point vous *recommandez* à \TeX d'insérer un saut de page à l'endroit où la commande apparaît (0 étant une faible recommandation et 4, une forte).

La commande `\enlargethispage`, comme son nom l'indique, permet d'allonger une page de $\langle longueur \rangle$ pour y faire tenir plus de texte. C'est une commande particulièrement utile pour éviter que la toute dernière ligne d'un chapitre ou d'un document se retrouve seule sur une page.

10.1.2 Coupure de mots

La coupure automatique des mots en fin de ligne est toujours active avec \TeX . C'est d'ailleurs pourquoi il est important d'indiquer à \TeX , à l'aide du paquetage **babel**, la langue d'un texte lorsque ce n'est pas l'anglais.

Il existe deux façons de contrôler la coupure de mots. La première, principalement utilisée lorsque \TeX refuse de couper un mot en fin de ligne, consiste à insérer des *suggestions* d'endroits où couper le mot avec la commande `\-`. Par exemple, en écrivant `vrai\sem\blance`, vous indiquez à \TeX qu'il est possible de diviser le mot en *vrai-semblance* ou *vraisem-blance*.

La seconde méthode, celle-là surtout utilisée lorsque \TeX ne reconnaît pas des mots qui reviennent souvent dans le document, consiste à fournir dans le préambule une liste d'exceptions avec la commande

```
\hyphenation{⟨liste⟩}
```

La $\langle liste \rangle$ est une suite de mots, séparés par des virgules, des blancs ou des retours à la ligne, dans lesquels les points de coupure sont identifiés par un trait d'union.

Exemple 10.2. La commande suivante, insérée dans le préambule, permet d'ajouter des points de coupure aux mots « puisque », « constante » et « vraisemblance » pour l'ensemble du document.

```
\hyphenation{puis-que,cons-tante,vrai-sem-blance}
```

□





Règle générale, garder les opérations d’ajustements de la mise en page — position des éléments flottants, sauts de page, lignes trop longues, etc. — pour la toute fin de la rédaction.

10.2 Au-delà de la police Computer Modern

Les documents \LaTeX standards sont facilement reconnaissables par leur police par défaut Computer Modern — celle utilisée dans ce paragraphe. Pour qui souhaitait briser la relative monotonie induite par cette uniformité, il a longtemps été difficile d’utiliser une autre police. Fort heureusement, la situation a beaucoup évolué et il est aujourd’hui assez simple de produire des documents \LaTeX utilisant des polices variées.


Avant d’aller plus loin, une mise en garde : si un document contient plus que quelques équations mathématiques très simples, le choix de police devient très restreint. En effet, peu de polices comprennent des symboles mathématiques et les informations nécessaires pour les assembler selon les hauts standards de qualité usuels de \LaTeX .

Cela dit, si vous souhaitez aller au-delà de la police Computer Modern sans trop vous compliquer la vie, il existe deux solutions principales.

1. Utiliser une police OpenType ou TrueType présente sur son système avec le moteur X_{\LaTeX} . C’est aujourd’hui la meilleure solution. La gestion des polices avec X_{\LaTeX} repose sur le paquetage standard **fontspec** ; consultez sa [documentation](#) .
2. Utiliser l’une ou l’autre des polices PostScript standards convenant pour du texte normal (Times, Palatino, Bookman, Charter, New Century Schoolbook, Helvetica). C’est la solution à privilégier si vous utilisez toujours $\pdf\LaTeX$. Il suffit alors de charger le paquetage approprié. Consultez à ce sujet la [documentation](#)  de l’ensemble de paquetages PSNFSS.

📖 fontspec















📖 psnfss2e


Le texte principal du présent document est en **Lucida Bright OT** , une police commerciale de très haute qualité offrant également un excellent support pour les mathématiques. Ses auteurs ont toujours été proches de la communauté \LaTeX . La Bibliothèque de l’Université Laval détient une licence d’utilisation de cette police. Les étudiants et le personnel de l’Université peuvent s’en procurer une copie gratuitement en écrivant à lucida@bibl.ulaval.ca.

10.3 Couleurs

L’ajout de couleur dans un document \LaTeX requiert le paquetage **xcolor** (Kern, 2022). Celui-ci définit d’abord plusieurs dizaines de couleurs que vous

TAB. 10.1 - Couleurs toujours disponibles lorsque le paquetage **xcolor** est chargé

 black	 blue	 brown	 cyan	 darkgray
 gray	 green	 lightgray	 lime	 magenta
 olive	 orange	 pink	 purple	 red
 teal	 violet	 white	 yellow	

pouvez ensuite utiliser directement dans votre document ; consultez la [documentation](#)  pour en connaître les différentes listes. Le [tableau 10.1](#) fournit la liste des couleurs toujours disponibles.

 **xcolor**

Un peu comme pour les changements d'attributs de police, il existe deux commandes pour modifier la couleur du texte :

```
\color{<nom>}
\textcolor{<nom>}{<texte>}
```

La première modifie la couleur de tout ce qui suit (à moins d'en limiter la portée avec des accolades) et la seconde, seulement pour *<texte>*. Dans les deux cas, *<nom>* est le nom d'une couleur.

```
texte {\color{red} en rouge}
et \textcolor{blue}{en bleu}      texte en rouge et en bleu
```





La commande `\definecolor` permet de définir de nouvelles couleurs selon plusieurs systèmes de codage. Le plus usuel demeure *Rouge, vert, bleu* (RVB ou RGB, en anglais), dans lequel une couleur est représentée par une combinaison de teintes — exprimées par un nombre entre 0 et 1 — de rouge, de vert et de bleu. Dans ce cas, la syntaxe de `\definecolor` est

```
\definecolor{<nom>}{rgb}{<valeur_r,valeur_v,valeur_b>}
```

où *<valeur_r>*, *<valeur_v>* et *<valeur_b>* sont respectivement les teintes de rouge, de vert et de bleu.

Exemple 10.3. La commande

```
\definecolor{acier}{rgb}{0.3,0.4,0.6}
```

définit une nouvelle couleur nommée **acier** composée de rouge 30 %, de vert 40 % et de bleu 60 % :  +  +  = . Une fois la couleur **acier** définie dans le préambule, elle devient disponible pour être utilisée dans les commandes `\color` et `\textcolor`. □

La commande `\colorlet`, dont la syntaxe simplifiée est

```
\colorlet{<nom>}{<couleur>}
```

permet de faire référence à la `<couleur>` déjà existante par `<nom>`. C'est pratique pour assigner un nom sémantique à une couleur.

10.4 Hyperliens et métadonnées de documents PDF

J'en ai déjà traité à quelques reprises, notamment à la [section 3.4](#) : le paquetage **hyperref** (Rahtz et Oberdiek, 2023) permet de transformer les références dans le texte en hyperliens cliquables lorsque le document est produit avec pdfL^AT_EX ou XeL^AT_EX. C'est très pratique lors de la consultation électronique d'un document.

Le paquetage offre une multitude d'options de configuration; je n'en présente que quelques unes. On accède aux options de configuration de **hyperref** via la commande `\hypersetup` dans le préambule. Celle-ci prend en arguments des paires `option=valeur` séparées par des virgules.

Une des principales choses que vous pourrez souhaiter configurer dans **hyperref**, c'est le comportement et la couleur des divers types d'hyperliens. Voici les options de configuration pertinentes, leur valeur (avec en gras la valeur par défaut) ainsi qu'une brève explication de chacune.

<code>colorlinks</code>	<code>true false</code>	colorer les liens selon leur type
<code>linktocpage</code>	<code>true false</code>	faire du numéro de page plutôt que du titre l'hyperlien dans la table des matières
<code>linkcolor</code>	<code><couleur></code>	couleur des liens internes
<code>urlcolor</code>	<code><couleur></code>	couleur des adresses URL
<code>citecolor</code>	<code><couleur></code>	couleur des citations
<code>allcolor</code>	<code><couleur></code>	couleur pour tous les types d'hyperliens

Les fichiers PDF peuvent contenir diverses métadonnées sur leur contenu. Le paquetage **hyperref** permet d'en définir certaines, notamment le titre ou l'auteur du document.

<code>pdftitle</code>	texte	titre du document PDF
<code>pdfauthor</code>	texte	auteur du document PDF

■ hyperref

Consultez la [documentation](#) de **hyperref** pour obtenir la liste complète des options de configuration des métadonnées.

Exemple 10.4. Le préambule du présent document contient les définitions de couleurs et d'options de **hyperref** suivantes :

```
\definecolor{link}{rgb}{0,0.4,0.6} % liens internes
\definecolor{url}{rgb}{0.6,0,0} % liens externes
\definecolor{citation}{rgb}{0,0.5,0} % citations
\hypersetup{%
  pdfauthor = {Vincent Goulet},
  pdftitle = {Rédaction avec LaTeX},
  colorlinks = true,
  linktocpage = true,
  urlcolor = url,
  linkcolor = link,
  citecolor = citation}
```

□



L'interaction du paquetage **hyperref** avec les autres est souvent délicate. Pour cette raison, il est fortement recommandé qu'il soit le tout dernier paquetage chargé dans le préambule.

10.5 Présentation de code informatique

L'environnement standard verbatim de \LaTeX permet de présenter du texte tel qu'il est entré dans le code source du document. Le texte qu'il contient est composé en police non proportionnelle et sa disposition exacte est respectée.

```
\begin{verbatim}
/* Hello World en C */
#include <stdio.h>

int main()
{
  printf("Hello world\n");
  return 0;
}
\end{verbatim}
/* Hello World en C */
#include <stdio.h>

int main()
{
  printf("Hello world\n");
  return 0;
}
```

listings

Si un document doit contenir beaucoup de code informatique et que vous souhaitez exercer un fin contrôle sur sa disposition et sa mise en forme, il vaut mieux vous tourner vers un paquetage spécialisé comme **listings** (Heinz et collab., 2023). La [documentation](#) du paquetage compare ses fonctionnalités à celles de plusieurs autres paquetages similaires.

Le paquetage **listings** peut effectuer automatiquement le marquage des mots-clés de nombreux langages de programmation, ajouter des numéros de ligne, importer du code de fichiers externes ou même indexer les mots-clés des extraits de code. À titre d'illustration, l'environnement `lstlisting` pourrait composer ainsi l'extrait de code C ci-dessus :

```
\begin{lstlisting}
/* Hello World en C */
#include <stdio.h>

int main()
{
    printf("Hello world\n");
    return 0;
}
\end{lstlisting}
```

```
1  /* Hello World en C */
2  #include <stdio.h>
3
4  int main()
5  {
6      printf("Hello world\n");
7      return 0;
8  }
```

Il serait trop long et nettement hors de la portée du présent ouvrage d'expliquer les nombreuses fonctionnalités de **listings**. Je précise simplement que c'est à l'aide du paquetage que j'ai composé les extraits de code et construit une grande partie de l'index.

Exemple 10.5. Pour parvenir à la présentation des extraits de code source \LaTeX de ce document, le paquetage **listings** est configuré dans le préambule de la manière suivante :

```
%% Couleurs
\definecolor{comments}{rgb}{0.5,0.55,0.6}

%% Configuration de listings
\lstset{language=[LaTeX]TeX,
        basicstyle=\ttfamily\NoAutoSpacing,
        keywordstyle=\mdseries,
        commentstyle=\color{comments}\slshape,
        extendedchars=true,
        showstringspaces=false,
```

```
backgroundcolor=\color{LightYellow1},
frame=leftline,
framerule=2pt,
framesep=5pt,
xleftmargin=7.4pt}
```

(La couleur `LightYellow1` est définie par `xcolor` lorsque le paquetage est chargé avec l'option `x11names`.) \square

10.6 Production de rapports avec l'analyse intégrée

Les publications scientifiques reposent souvent sur une forme ou une autre d'analyse numérique ou statistique, la production de code informatique, une simulation stochastique, etc. La portion développement et analyse est alors produite avec un certain outil et la publication, avec un outil d'édition séparé — \LaTeX dans le cas qui nous occupe. Or, tous les auteurs ont vécu cette situation : les résultats de l'analyse changent et il faut modifier le rapport en conséquence, refaire les tableaux et les graphiques, retracer cette valeur isolée au fil du texte directement tirée de l'analyse... Seule la quantité de temps perdu rivalise avec le risque d'erreur.

Il existe pourtant une meilleure façon de travailler.

Cette meilleure façon de faire, tirée du concept de *programmation lettrée*, consiste à combiner dans un seul et même document l'analyse et le rapport, puis de produire automatiquement une partie du second à partir de la première.

Les utilisateurs du système statistique R bénéficient d'une mise en œuvre simple et élégante du concept ci-dessus avec l'outil Sweave (Leisch, 2002). Un fichier Sweave est à la base un document \LaTeX dans lequel on a inséré du code R à l'intérieur de balises spéciales `<<>=` et `@` tirée de la syntaxe `noweb` (Ramsey, 2011), comme ceci :

```
\section{Commandes R}

L'utilisateur de R interagit avec l'interprète en entrant
des commandes à l'invite de commande:
<<>=
2 + 3
pi
cos(pi/4)
@
La commande \verb=exp(1)= donne \Sexpr{exp(1)},
```

la valeur du nombre e .

Par convention, on enregistre un tel document sous un nom se terminant par l'extension `.Rnw`. Sa compilation s'effectue en deux étapes :

1. Le fichier `.Rnw` est passé à la commande `Sweave()` de R. Celle-ci retrace les extraits de code et les remplace par des environnements \LaTeX contenant, par défaut, les expressions R et leur résultat dans des environnements `Sinput` et `Soutput`. Elle évalue également les expressions R se trouvant dans les commandes `\Sexpr` pour les remplacer par leur résultat. Cela produit un fichier `.tex` :

```
\section{Commandes R}

L'utilisateur de R interagit avec l'interprète en entrant
des commandes à l'invite de commande:
\begin{Schunk}
\begin{Sinput}
> 2 + 3
\end{Sinput}
\begin{Soutput}
[1] 5
\end{Soutput}
\begin{Sinput}
> pi
\end{Sinput}
\begin{Soutput}
[1] 3.141593
\end{Soutput}
\begin{Sinput}
> cos(pi/4)
\end{Sinput}
\begin{Soutput}
[1] 0.7071068
\end{Soutput}
\end{Schunk}
La commande \verb=exp(1)= donne 2.71828182845905,
la valeur du nombre  $e$ .
```

2. Le fichier `.tex` est compilé comme d'habitude.

`Sweave` se révèle particulièrement utile pour créer des graphiques à partir de R : tout ce que l'on doit conserver dans son fichier `.Rnw`, c'est le code

pour créer le graphique. Il est également possible de contrôler l'exécution des blocs de code et l'affichage du code source et des résultats par le biais d'options placées à l'intérieur de la balise d'ouverture $\langle\langle\rangle\rangle=$.

Vous pouvez consulter la documentation de Sweave en entrant dans R la commande

```
| > vignette("Sweave", package = "utils")
```

Pweave [↗](#) est un système similaire à Sweave pour le langage Python.

Inspiré de Sweave, knitr ([Xie, 2023](#)) permet également d'entrelacer du code \LaTeX et du code R. Cet outil offre plus d'options de traitement que Sweave, mais au prix d'une complexité accrue.

En terminant, je souligne que Sweave n'est qu'un exemple de système de programmation lettrée. Il en existe plusieurs autres. Je vous invite à identifier celui qui correspond le mieux à vos besoins.



On doit le concept de programmation lettrée au créateur de \TeX , Donald Knuth. En fait, tout le code source de \TeX est écrit en programmation lettrée ! La [page Wikipedia](#) [↗](#) consacrée au sujet offre un très bon survol de l'historique et de la nature du concept.

10.7 Diapositives

Il n'est pas rare qu'une publication scientifique fasse l'objet d'une présentation dans le cadre d'un colloque ou d'un séminaire. Lorsque le texte a été rédigé avec \LaTeX , il est tout naturel de souhaiter le réutiliser pour la préparation de diapositives — surtout si le texte comporte de nombreuses équations mathématiques qu'il serait extrêmement long de retranscrire dans un logiciel de présentation comme PowerPoint.

Fort heureusement, il est tout à fait possible de composer ses diapositives avec \LaTeX . La classe standard slides produit des diapositives élégantes, quoique minimalistes. Elles vous conviendront si vous ne recherchez rien de plus que du texte noir sur fond blanc.

Cependant, l'outil devenu le standard *de facto* pour la production de diapositives est la classe beamer ([Tantau et collab., 2023](#)). Celle-ci compte un grand nombre de thèmes et de gabarits élaborés, elle rend très simple l'insertion d'animations d'une diapositive à l'autre et... elle produit des diapositives en couleur. Consultez la [documentation](#) [↗](#) de beamer pour apprendre à utiliser la classe. Produire des diapositives de grande qualité avec les gabarits fournis avec la classe se révèle simple et rapide.



Quelle que soit la classe utilisée, les diapositives produites avec \LaTeX se présentent sous forme de fichier PDF que l'on les projette avec une liseuse PDF en mode plein écran.

10.8 Gestion des versions et travail collaboratif

Plusieurs personnes travaillent sur un même fichier, ou encore, une seule personne y travaille, mais de plusieurs postes de travail différents. Quelle est la plus récente version du fichier ? Un ajout fait hier dans le fichier a-t-il été pris en compte par une collègue aujourd'hui ? Une modification apportée au fichier n'est plus nécessaire ; comment retourner en arrière facilement ?

Les informaticiens ont résolu ce genre de problèmes il y a des dizaines d'années avec les systèmes de gestion de versions. Le système le plus populaire dans le monde en ce moment est Git ([Chacon et Straub, 2014](#)) (que l'on prononce avec un « g » dur, comme dans « guitare »).

Bien que développés à l'origine pour la gestion du code source de logiciels, les systèmes de gestion de versions conviennent parfaitement pour les sources \LaTeX . L'utilisation d'un tel système permet de :

- ▶ toujours savoir quelle est la plus récente version d'un fichier ;
- ▶ travailler à plusieurs personnes simultanément sur un même fichier ;
- ▶ revenir aisément à une version antérieure d'un fichier ;
- ▶ comparer deux versions d'un fichier pour connaître les modifications qui y ont été apportées ;
- ▶ gérer automatiquement les éventuels conflits de modification d'un fichier ;
- ▶ disposer en tout temps d'une copie de secours de son travail lorsque l'on a recours à un dépôt centralisé.

Un système de gestion de versions est un outil qui vous permettra d'augmenter considérablement votre productivité ou celle de votre équipe de travail au moment de rédiger un ouvrage scientifique.



Le code source de cet ouvrage présente est conservé dans un dépôt Git public duquel vous pouvez en tout temps obtenir la plus récente version. L'adresse URL du dépôt se trouve dans la page des notices de copyright au début du document.




10.9 Exercices

Pour les exercices 10.1–10.5, utiliser le fichier `exercice-trucs.tex`. Celui-ci reprend une partie de la documentation de la classe `ulthese` pour les thèses et mémoires de l'Université Laval. Avant toute chose, passez rapidement en revue le préambule du document. Vous constaterez que le document :

- ▶ est compatible tant avec `pdfLATEX` qu'avec `XYLATEX` grâce à un test sur le moteur utilisé effectué dans le préambule;;
- ▶ définit des nouvelles commandes `\class` et `\fichier` pour composer, respectivement, les noms de classes et les noms de fichiers;
- ▶ utilise la commande `\doc` de l'[exemple 9.5](#);
- ▶ charge le paquetage **hyperref**, ce qui transforme les titres de la table des matières, les renvois aux notes de bas de page et les liens externes en hyperliens.

10.1 Compiler et visualiser le fichier sans aucune modification. Le texte est composé dans la police par défaut Computer Modern.

Ensuite, modifier le préambule du document pour composer le document avec les polices suivantes :

- ▶ **STIX**  (version 2) pour le texte romain;
- ▶ **Fira Sans**  pour le texte sans empattements;
- ▶ **Fira Mono**  pour le texte à largeur fixe.

Avec `XYLATEX`, insérer les lignes suivantes après le chargement du paquetage **fontspec** :

```
\setmainfont{STIXTwoText}
[
  Extension = .otf,
  UprightFont = *-Regular,
  BoldFont = *-SemiBold,
  ItalicFont = *-Italic,
  BoldItalicFont = *-SemiBoldItalic,
]
\setsansfont{FiraSans}
[
  Extension = .otf,
  UprightFont = *-Regular,
  BoldFont = *-SemiBold,
```

```

    ItalicFont = *-Italic,
    BoldItalicFont = *-SemiBoldItalic,
    Scale = 0.92
]
\setmonofont{FiraMono}
[
    Extension = .otf,
    UprightFont = *-Regular,
    BoldFont = *-Medium,
    ItalicFont = *-Oblique,
    BoldItalicFont = *-MediumOblique,
    Scale = 0.90
]

```

Avec pdfL^AT_EX, insérer les lignes suivantes après le chargement du paquetage **fontenc** :

```

\usepackage{stix2}
\usepackage[scale=0.92]{FiraSans}
\usepackage[scale=0.90]{FiraMono}

```

- 10.2** Configurer le paquetage **hyperref** pour que les hyperliens dans la table des matières soient ancrés aux numéros de page plutôt qu’aux titres de section.
- 10.3** Charger le paquetage **xcolor** et ajouter l’option `colorlinks` à **hyperref**, puis recompiler le document. Observer les changements.
- 10.4** En s’inspirant de l’[exemple 10.4](#), modifier la couleur des liens internes et externes.
- 10.5** Charger le paquetage **listings** et modifier l’environnement `verbatim` que l’on trouve dans le document pour un environnement `lstlisting`. En s’inspirant de l’[exemple 10.5](#), configurer la présentation des extraits de code pour utiliser une police non proportionnelle (`\ttfamily`) et un arrière-plan de la couleur standard `lightgray`.

A Classe pour les thèses et mémoires de l'Université Laval

La classe `ulthese` (Goulet, 2018) permet de composer des thèses et mémoires immédiatement conformes aux règles générales de présentation matérielle de la Faculté des études supérieures et postdoctorales de l'Université Laval. Ces règles définissent principalement la présentation des pages de titre des thèses et mémoires ainsi que la disposition du texte sur la page. La classe est compatible tant avec \LaTeX qu'avec X_{\LaTeX} .


La classe `ulthese` est basée sur la classe `memoir`. L'intégralité des fonctionnalités de `memoir` se retrouve donc disponible dans `ulthese`. Les deux classes sont livrées avec la distribution \TeX Live.

Outre `memoir`, la classe charge par défaut un certain nombre de paquets, notamment les essentiels pour la rédaction en français. Il n'est donc pas nécessaire de charger de nouveau les paquets suivants : **`babel`**, **`numprint`**, **`natbib`**, **`fontspec`** (moteur X_{\LaTeX} seulement), **`graphicx`**, **`xcolor`**, **`textcomp`**. Le paquetage **`geometry`** est incompatible avec la classe à cause de sa mauvaise interaction avec `memoir`. Son chargement dans le préambule du document cause une erreur lors de la compilation.

Afin de faciliter la rédaction, la classe est livrée avec un ensemble de gabarits sur lesquels se baser pour :

- ▶ les fichiers maîtres de divers types de thèses et mémoires (standard, sur mesure, en cotutelle, en bidiplomation, en extension, etc.);
- ▶ les fichiers des parties les plus usuelles (résumés français et anglais, avant-propos, introduction, chapitres, conclusion, etc.).

Ces gabarits comportent des commentaires succincts pour vous guider dans la préparation de votre ouvrage. La [documentation](#) de la classe explique en détail le contenu des gabarits.

 `ulthese`

B Solutions des exercices

Chapitre 2

2.2 Il y a trois modifications principales à apporter au fichier.

1. La commande `\LaTeX` qui imprime le logo de LaTeX n'ayant pas d'argument, il faut l'écrire sous la forme `{\LaTeX}` ou `\LaTeX{}` pour éviter que l'espace qui suit ne soit avalée par la commande.
2. Il faut délimiter par des accolades la zone à laquelle la commande `\bfseries` doit s'appliquer :

```
| {\bfseries limitée}
```

3. Il faut utiliser l'environnement `enumerate` pour les listes numérotées. L'environnement `itemize` sert pour les listes à puces.

Chapitre 5

5.1 Une première boîte verticale de 10 cm de large contient le texte :

```
| \parbox{10cm}{Ce bloc [...] la ligne.}
```

Cette boîte peut être placée dans une boîte horizontale encadrée avec `\fbox`. Celle-ci peut à son tour être placée dans une autre boîte horizontale encadrée, de manière à obtenir un cadrage double. Pour centrer le tout sur la ligne, il reste à placer le tout dans un environnement `center` :

```
| \begin{center}  
|   \fbox{\fbox{\parbox{10cm}{Ce bloc [...] la ligne.}}}  
| \end{center}
```

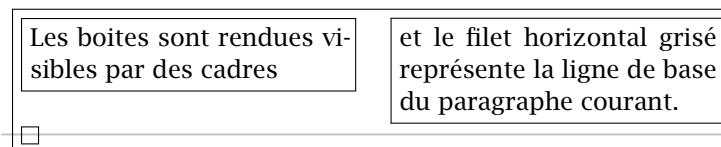
5.2 L'idée consiste à créer une seconde ligne dans la `mini page` externe sans que celle-ci n'occupe aucun espace. Pour ce faire, on insère du contenu vide avec `\mbox{}`, tel qu'expliqué à la [page 54](#). Le code

```

\begin{minipage}[b]{...}
  \parbox[t]{...}{...} \hfill \parbox[t]{...}{...} \\
  \mbox{}
\end{minipage}

```

produit donc le résultat voulu :



(Sans le cadre, la boîte de la seconde ligne n'occupe aucun espace.)

- 5.3** La solution la plus simple consiste à réunir les deux premières boîtes dans une `minipage` dans laquelle les deux boîtes seront alignées tel que désiré, puis à aligner la `minipage` avec la troisième boîte. Cependant, il faut insérer une seconde ligne invisible dans la `minipage` afin de pouvoir l'aligner par le bas avec la boîte de droite :

```

\begin{minipage}[b]{80mm}
  \parbox[t]{30mm}{...} \hfill \parbox[t]{45mm}{...} \\
  \mbox{}
\end{minipage}
\hfill
\parbox[b]{35mm}{...}

```

Chapitre 6

- 6.1** Les paquets `babel` et `numprint` étant chargés dans le fichier de gabarit, le code pour créer le tableau est le suivant :

```

\begin{tabular}{lp{7.5cm}r}
  \toprule
  {\No} lot & Description & Prix (\$) \\
  \midrule
  U-236 & Ordinateur [...] & 998 \\
  U-374 & Chaise [...] & 275 \\
  U-588 & Table [...] & \nombre{1125} \\
  \bottomrule
\end{tabular}

```

6.2 Pour effectuer les modifications demandées, il faut :

- i) utiliser la commande `\multicolumn` dans l'entête du tableau pour centrer le titre de la deuxième colonne sans autrement centrer le contenu de la colonne;
- ii) remplacer l'environnement `tabular` par l'environnement `tabularx` de memoir, spécifier une largeur de tableau `\textwidth`, changer le format de la deuxième colonne pour X afin que la largeur de celle-ci s'ajuste automatiquement pour combler celle du tableau.

```
\begin{tabularx}{\textwidth}{lXr}
\toprule
{\No} lot & \multicolumn{1}{c}{Description}
& Prix (\$) \\
\midrule
U-236 & Ordinateur [...] & 998 \\
U-374 & Chaise [...] & 275 \\
U-588 & Table [...] & \nombre{1125} \\
\bottomrule
\end{tabularx}
```

6.3 Tout d'abord, vous remarquerez que la commande

```
\newsubfloat{table}
```

est déjà présente dans le préambule du fichier. Si l'on souhaite placer des sous-légendes au-dessus de chacun des deux tableaux, le code du tableau devient :

```
\begin{table}
\caption{Conversion du nombre décimal $23,31$
en binaire.}
\begin{minipage}[t]{0.45\linewidth}
\subcaption{\<texte\> % ajout
\begin{tabular*}{\linewidth}{crrcc}
...
\end{tabular*}
\end{minipage}
\hfill
\begin{minipage}[t]{0.45\linewidth}
\subcaption{\<texte\> % ajout
\begin{tabular*}{\linewidth}{ccccc}
...
\end{tabular*}
\end{minipage}
```

```
\end{tabular*}
\end{minipage}
\end{table}
```

6.4 Le préambule du document devrait contenir la déclaration

```
\usepackage{pdfpages}
```

pour charger le paquetage **pdfpages**. Ensuite, à l'endroit où l'on souhaite insérer la couverture du présent document dans le document, il s'agit de placer la commande

```
\includepdf[pages=1]{formation-latex-ul}
```

Chapitre 7

7.1 La commande pour produire le symbole \circ se trouve dans le [tableau 7.5](#). Il est possible de produire les symboles de dérivée ' avec la commande `\prime` ([tableau 7.8](#)) ou simplement avec le caractère '.

La dérivée de la fonction composée $f \circ g(x) = f[g(x)]$ est $\{f[g(x)]\}' = f'[g(x)] g'(x)$. La dérivée seconde du produit des fonctions f et g est

```
\begin{equation*}
[f(x) g(x)]'' =
f''(x) g(x) + 2 f'(x) g'(x) +
f(x) g''(x).
\end{equation*}
```

7.2 Les lignes de l'équation sont alignées juste à droite du premier symbole + à la première ligne. Remarquez l'usage des commandes `\cdots` et `\dots` dans la dernière ligne : \TeX choisit correctement la position centrée des points entre les opérateurs d'addition, mais pas entre les accolades.

```
\begin{align*}
f(x + h, y + k) &= f(x, y) + \\
&\left\{ \frac{\partial f(x,y)}{\partial x} h + \right. \\
&\left. \frac{\partial f(x,y)}{\partial y} k \right\} \\
&\quad \&
```

```

\frac{1}{2}
\left\{
\frac{\partial^2 f(x,y)}{\partial x^2} h^2 +
\frac{\partial^2 f(x,y)}{\partial x \partial y} kh +
\frac{\partial^2 f(x,y)}{\partial y^2} k^2
\right\} \setminus
&+
\frac{1}{6} \setminus\cdots\setminus + \dots +
\frac{1}{n!} \setminus\cdots\setminus + R_n.
\end{align*}

```

7.3 Il faut utiliser `\dfrac` pour obtenir des fractions dans une branche de cases de la même taille que dans une équation hors paragraphe :

```

\begin{equation*}
\tilde{f}(x) =
\begin{cases}
0, & x \leq c_0 \\
\dfrac{F_n(c_j) - F_n(c_{j-1})}{c_j - c_{j-1}} =
\dfrac{n_j}{n(c_j - c_{j-1})}, &
c_{j-1} < x \leq c_j \\
0, & x > c_r.
\end{cases}
\end{equation*}

```

Chapitre 8

8.1 L'entrée sur [BIB_TE_X](#) dans Wikipedia fournit la liste des champs obligatoires et optionnels pour chaque type d'entrée. La clé est laissée vacante dans les réponses ci-dessous.

a) Cette entrée utilise le type `article`.

```

@article{
  author = {Frank Mittelbach},
  title = {How to Influence the Position of Float
           Environments Like Figure and Table In
           {\LaTeX}?},
  journal = {{TUG}boat},
  year = 2014,
  volume = 35,

```

```
number = 3,
pages = {258-254},
language = {english}}
```

b) Cette entrée utilise le type book.

```
@book{,
author = {Peter Wilson},
title = {The Memoir Class for Configurable
Typesetting},
publisher = {The Herries Press},
year = 2013,
edition = 8,
note = {Maintained by Lars Madsen},
url = {https://www.ctan.org/pkg/memoir},
language = {english}}
```

c) La réponse ci-dessous contient les prénoms des auteurs, simplement afin d'illustrer que $\text{BIB}_{\text{T}}\text{X}$ est en mesure de les abrégier. Des accolades { } sont utilisées dans le titre pour préserver la casse de « PSTricks » et de « PostScript ».

```
@manual{,
author = {Timothy {V}an Zandt and Denis Girou and
Herbert Vo{\ss}},
title = {{PSTricks} --- {PostScript} Macros for
Generic {\TeX}},
year = 2014,
url = {https://www.ctan.org/pkg/pstricks-base},
language = {english}}
```

Chapitre 9

- 9.1 a) `\newcommand{\NASA}{\textsc{nasa}}`
 b) `\newcommand{\sigle}[1]{\textsc{\MakeLowercase{#1}}}`
 c) `\newcommand{\sigle}[1]{\MakeUppercase{#1}}`

9.2 Avec la définition

```
\newcommand{\doc}[3][documentation]{%
\href{#3}{#1~\raisebox{-0.2ex}{\faExternalLink}}%
\marginpar{\faBookmark~\texttt{#2}}}
```


la déclaration à deux arguments

```
| \doc{hyperref}{https://texdoc.net/pkg/hyperref}
```

produit toujours [documentation](#) .

hyperref

9.3 La définition suivante donne les résultats demandés :

```
| \newenvironment{citation}
|   {\begin{quote}
|     \begin{tabularx}{\linewidth}{X}
|       \toprule\small\sffamily}%
|     \\ \bottomrule
|     \end{tabularx}
|   \end{quote}}
```

Dans le troisième argument, la présence de la commande de fin de ligne `\\` doit précéder `\bottomrule`.

Chapitre 10

10.2 Tel qu'expliqué à la [section 10.4](#), il faut insérer dans le préambule du document la commande

```
| \hypersetup{linktocpage=true}
```

ou, plus simplement,

```
| \hypersetup{linktocpage}
```

10.3 En conservant l'ajout de l'exercice précédent, le préambule doit contenir la commande

```
| \hypersetup{colorlinks, linktocpage}
```

Les hyperliens apparaitront alors en couleur selon les paramètres par défaut de **hyperref**.

10.4 Il est possible soit d'utiliser des couleurs prédéfinies de **xcolor** ([tableau 10.1](#)), soit d'en définir de nouvelles avec `\definicolor`. Les couleurs sont ensuite sélectionnées via les options `linkcolor` (liens internes) et `urlcolor` (liens externes) de **hyperref**.

Exemple utilisant des couleurs prédéfinies :

```
| \hypersetup{colorlinks, linktocpage,
|   linkcolor=brown, urlcolor=blue}
```

Exemple avec de nouvelles couleurs :

```
\definecolor{link}{rgb}{0,0.4,0.6}
\definecolor{url}{rgb}{0.6,0,0}
\hypersetup{colorlinks, linktocpage,
  urlcolor=url, linkcolor=link}
```

- 10.5** S'il y avait plusieurs extraits de code dans le document, mieux vaudrait les configurer tous à l'identique dans le préambule du document avec

```
\lstset{basicstyle=\ttfamily,
  backgroundcolor=\color{lightgray}}
```

Ensuite,

```
\begin{lstlisting}
latex ulthese.ins
\end{lstlisting}
```

donne le résultat demandé.

Pour un seul extrait, il est également possible de simplement charger le paquetage dans le préambule et d'effectuer la configuration à l'ouverture de l'environnement, comme ceci :

```
\begin{lstlisting}[basicstyle=\ttfamily,
  backgroundcolor=\color{lightgray}]
latex ulthese.ins
\end{lstlisting}
```

Bibliographie

- Beccari, C. 1997, « Typesetting mathematics for science and technology according to ISO 31/XI », *TUGboat*, vol. 18, n° 1, p. 39-48, <https://tug.org/TUGboat/tb18-1/tb54becc.pdf>.
- Beebe, N. H. F. 2004, « 25 years of T_EX and METAFONT: Looking back and looking forward », *TUGboat*, vol. 25, n° 1, p. 7-30, <https://www.tug.org/TUGboat/tb25-1/beebe-2003keynote.pdf>. TUG 2003 Keynote Address.
- Bezoz, J. 2019, *Customizing lists with the **enumitem** package*, <https://www.ctan.org/pkg/enumitem>.
- Braams, J. et J. Bezoz. 2023, *Babel*, <https://www.ctan.org/pkg/babel>.
- Carlisle, D. 2014, *The **bm** Package*, <https://www.ctan.org/pkg/bm>.
- Carlisle, D. et The L^AT_EX₃ Project. 2021, *Packages in the 'graphics' Bundle*, <https://www.ctan.org/pkg/graphics>.
- Chacon, S. et B. Straub. 2014, *Pro Git*, 2^e éd., Apress, <https://git-scm.com/book/fr/v2>. Version en ligne.
- Daly, P. W. 2010, *Natural Sciences Citations and References*, <https://www.ctan.org/pkg/natbib>.
- Fairbairns, R. 2022, ***footmisc** — a portmanteau package for customising footnotes in L^AT_EX*, <https://www.ctan.org/pkg/footmisc>.
- Fear, S. 2020, *Publication quality tables in L^AT_EX*, <https://www.ctan.org/pkg/booktabs>.
- Goulet, V. 2018, *ulthese : la classe pour les thèses et mémoires de l'Université Laval*, Faculté des études supérieures et doctorales, Université Laval, <https://www.ctan.org/pkg/ulthese>.

- Goulet, V. 2023, « Styles bibliographiques français et français », <https://www.ctan.org/pkg/francais-bst>.
- Harders, H. 2012, *The numprint package*, <https://www.ctan.org/pkg/numprint>.
- Heinz, C., B. Moses et J. Hoffmann. 2023, *The listings Package*, <https://www.ctan.org/pkg/listings>.
- Jeffrey, A. et F. Mittelbach. 2021, *inputenc.sty*, <https://www.ctan.org/pkg/inputenc>.
- Kern, D. U. 2022, *Extending L^AT_EX's color facilities: the xcolor package*, <https://www.ctan.org/pkg/xcolor>.
- Kline, P. et F. Charette. 2023, *biber — A Backend Bibliography Processor for Biblatex*, <https://www.ctan.org/pkg/biber>.
- Kopka, H. et P. W. Daly. 2003, *Guide to L^AT_EX*, 4^e éd., Addison-Wesley, ISBN 978-032117385-0.
- Krüger, M. 2022, *The fontawesome5 package*, <https://www.ctan.org/pkg/fontawesome5>.
- Lehman, P. 2023, *The Biblatex Package — Programmable Bibliographies and Citations*, <https://www.ctan.org/pkg/biblatex>.
- Lehman, P. et J. Wright. 2022, *The csquotes Package: Context Sensitive Quotation Facilities*, <https://www.ctan.org/pkg/csquotes>.
- Leisch, F. 2002, « Sweave: Dynamic generation of statistical reports using literate data analysis », dans *Compstat 2002 — Proceedings in Computational Statistics*, sous la direction de W. Härdle et B. Rönz, Physica Verlag, Heidelberg, p. 575-580, ISBN 3-79081517-9, <https://www.stat.uni-muenchen.de/~leisch/Sweave>.
- Malo, M. 1996, *Guide de la communication écrite au cégep, à l'université et en entreprise*, Québec Amérique.
- Matthias, A. 2022, *The pdfpages Package*, <https://www.ctan.org/pkg/pdfpages>.
- Mittelbach, F. 2014a, « How to influence the position of float environments like figure and table in L^AT_EX? », *TUGboat*, vol. 35, n^o 3, p. 258-254, <https://www.tug.org/TUGboat/tb35-3/tb111mitt-float.pdf>.

- Mittelbach, F. 2014b, « How to influence the position of float environments like figure and table in \LaTeX ? », *TUGboat*, vol. 35, n° 3, p. 258-254.
- Mittelbach, F., R. Fairbairns et W. Lemberg. 2016, *\LaTeX font encodings*, <https://www.ctan.org/pkg/encguide>.
- Mori, L. F. 2009, « Managing bibliographies with \LaTeX », *TUGboat*, vol. 30, n° 1, p. 36-48, <https://www.tug.org/TUGboat/tb30-1/tb94mori.pdf>.
- Pakin, S. 2021, *The Comprehensive \LaTeX Symbol List*, <https://www.ctan.org/pkg/comprehensive>.
- Patasnik, O. 1988, *BIB \TeX ing*, <https://www.ctan.org/pkg/bibtex>.
- Rabagliati, M. 1999, *Paul à la campagne*, La Pastèque, ISBN 978-2-92258501-8.
- Rahtz, S. et H. Oberdiek. 2023, *Hypertext marks in \LaTeX : a manual for hyperref*, <https://www.ctan.org/pkg/hyperref>.
- Ramsey, N. 2011, « Noweb — a simple, extensible tool for literate programming », <https://www.cs.tufts.edu/~nr/noweb>.
- Robertson, W. et K. Hosny. 2022, *The **fontspec** package: Font selection for X_{\LaTeX} and Lua \LaTeX* , <https://www.ctan.org/pkg/fontspec>.
- Robertson, W., P. Stephani et K. Hosny. 2020, *Experimental Unicode Mathematical Typesetting: The **unicode-math** Package*, <https://www.ctan.org/pkg/unicode-math>.
- Schmidt, W. 2002, *The **icomma** package for $\LaTeX_{2\epsilon}$* , <https://www.ctan.org/pkg/icomma>.
- American Mathematical Society. 2002, *User's Guide for the **amsmath** Package*, <https://www.ctan.org/pkg/amsmath>.
- Tantau, T. 2023, *PGF and TikZ — Graphic System for \TeX* , <https://www.ctan.org/pkg/pgf>.
- Tantau, T., J. Wright et V. Miletić. 2023, *The BEAMER Class*, <https://www.ctan.org/pkg/beamer>.
- Thurnherr, T. 2014, « \LaTeX document class options », *TUGboat*, vol. 35, n° 3, p. 245-247, <https://tug.org/TUGboat/tb35-3/tb111thurnherr.pdf>.

- Wikilivres. 2023, « LaTeX — wikilivres, la bibliothèque libre », <https://fr.wikibooks.org/wiki/LaTeX>.
- Wikipedia. 2023, « Mathematical operators and symbols in Unicode », https://en.wikipedia.org/wiki/Mathematical_operators_and_symbols_in_Unicode.
- Wilson, P. 2022, *The Memoir Class for Configurable Typesetting*, 8^e éd., The Herries Press, <https://www.ctan.org/pkg/memoir>. Maintained by Lars Madsen.
- Xie, Y. 2023, « **knitr**: A general-purpose package for dynamic report generation in R », <https://cran.r-project.org/package=knitr>. R package.
- Van Zandt, T., D. Girou et H. Voß. 2014, *PSTricks — PostScript Macros for Generic T_EX*, <https://www.ctan.org/pkg/pstricks-base>.

Index

Cet index contient des références aux commandes et environnements \LaTeX , ainsi qu'aux noms de paquetages et de classes. Le premier numéro indique habituellement, mais pas toujours, la page où un concept est introduit, défini ou expliqué.

- `\!`, 89
- `\#`, 17
- `\$`, 17
- `\%`, 17
- `\&`, 17
- `\,`, 86, 89
- `\-`, 131
- `\:`, 89
- `\;`, 89
- `\|`, 94
- `\`, 27, 48, 64, 65, 97, 102, 129, 130, 151
- `_`, 17

- `\above`, 14
- `abstract`, 30
- `\AE`, 18
- `\ae`, 18
- `align`, 95-98, 100-102, 104, 105, 148, 149
- `align*`, 95
- `aligned`, 99, 100
- `\alpha`, 92, 94, 95
- `amsmath`, 82, 83, 85-88, 90, 93-97, 100, 102, 104, 155
- `\angle`, 71, 77, 94
- `\approx`, 101
- `\caption`, 73
- `\arbox`, 56
- `\arccos`, 89
- `\arcsin`, 89
- `\arctan`, 89
- `\arg`, 89
- `article`, 10, 21, 23, 24, 26, 30
- `\ast`, 93
- `\author`, 27, 28
- `\autoref`, 36, 41, 76

- `babel`, 11, 17, 20-22, 47, 55, 75, 78, 112, 117, 121, 131, 143, 146
- `babel-french`, 20, 47
- `\backmatter`, 31
- `\backslash`, 94
- `\bar`, 94
- `beamer`, 139
- `\beta`, 92
- `\bfseries`, 13, 44
- `Biber`, 108

- biblatex, 108
- biblatex**, 108
- \bibliography, 116, 118
- \bibliographystyle, 113, 116, 118
- BI \TeX , 107
- \Big, 100, 101
- \big, 100, 101
- \Bigg, 100, 101
- \bigg, 100, 101
- \bm, 103, 104
- bm**, 103, 153
- Bmatrix, 106
- bmatrix, 104, 106
- book, 10, 21, 23-27, 31
- booktabs**, 62, 63, 65
- \bottomrule, 65, 66, 74, 76, 146, 147, 151
- \boxplus, 93
- \boxtimes, 93
- \cap, 93
- \caption, 75, 76, 79, 147
- cases, 100, 106, 149
- \cdot, 93
- \cdots, 87, 148, 149
- center, 13, 48, 145
- \centering, 48, 68, 74, 76
- \chapter, 30
- \chaptitelfont, 123
- \chi, 92
- \circ, 93, 148
- \circle, 68, 69
- \citation, 14, 16, 127, 135, 151
- citation, 127, 128
- \cite, 114
- \citeauthor, 115
- \citep, 115, 118
- \citet, 115, 116, 118
- \citeyear, 115
- classe
 - article, 10, 21, 23, 24, 26, 30
 - beamer, 139
 - book, 10, 21, 23-27, 31
 - letter, 10, 23
 - memoir, viii, ix, 26, 27, 29-31, 33, 41, 43, 45, 50, 51, 63, 65, 68, 74, 78, 123, 125, 130, 143, 147, 166
 - report, 10, 23-25, 27, 30
 - slides, 10, 23, 24, 139
 - ulthese, 121, 141, 143, 153
- \clearpage, 130
- \cleartorecto, 130
- \cleartoverso, 130
- \cmidrule, 65
- \color, 133, 136, 137, 152
- \colorlet, 134
- \cos, 89, 137, 138
- \cosh, 89
- \cot, 89
- \coth, 89
- \csc, 89
- csquotes**, 18, 154
- \cup, 93
- \dashbox, 69
- \date, 27
- \ddot, 94
- \ddots, 87
- \DeclareMathOperator, 90
- \definecolor, 133, 135, 136, 151, 152
- \deg, 89
- \Delta, 92
- \delta, 92
- description, 47
- \det, 89
- \dfrac, 85, 104-106, 149
- \dim, 89
- \displaybreak, 102

- `\div`, 93
- `\documentclass`, 9
- `\dots`, 87, 148, 149
- `\dotsb`, 88
- `\dotsc`, 88
- `\dotsi`, 88
- `\dotsm`, 88
- `\dotso`, 88
- `\Downarrow`, 94
- `\downarrow`, 94

- `\ell`, 94
- `\emph`, 12, 45, 51
- `\emptyset`, 94
- EndNote, 109
- `\enlargethispage`, 131
- enumerate, 46, 47
- enumitem**, 47, 153
- environnement
 - abstract, 30
 - align, 95-98, 100-102, 104, 105, 148, 149
 - align*, 95
 - aligned, 99, 100
 - Bmatrix, 106
 - bmatrix, 104, 106
 - cases, 100, 106, 149
 - center, 13, 48, 145
 - citation, 127, 128
 - description, 47
 - enumerate, 46, 47
 - equation, 82, 85, 90, 94-96, 148, 149
 - figure, 73, 77-79
 - gather, 96
 - gather*, 95
 - itemize, 46, 47
 - lstlisting, 136, 152
 - minipage, 56, 58, 59, 68, 77, 78, 130, 146-148
 - multline, 96
 - multline*, 95
 - NoHyper, 116
 - package, 139
 - picture, 67-69
 - pmatrix, 106
 - quotation, 48
 - quote, 13, 14, 48, 127, 151
 - Sinput, 138
 - Soutput, 138
 - split, 96, 97
 - table, 73, 74, 76, 79, 147, 148
 - tabular, 63-66, 74, 76, 146-148
 - tabular*, 63, 64
 - tabularx, 63, 64, 66, 147, 151
 - titlepage, 28
 - titlingpage, 28
 - verbatim, 50, 135
 - verbatim*, 50
 - Vmatrix, 106
 - vmatrix, 106

- `\epsilon`, 92
- `\eqref`, 97
- equation, 82, 85, 90, 94-96, 148, 149
- `\equiv`, 93
- `\eta`, 92
- `\exists`, 94
- `\exp`, 89, 137, 138

- `\fbox`, 54, 145
- figure, 73, 77-79
- fontawesome5**, 126, 154
- fontenc**, 20, 21, 142
- fontspec**, 19, 21, 28, 117, 132, 141, 143, 155
- footmisc**, 50, 153
- `\footnote`, 49
- `\footnotesize`, 45

- `\forall`, 94
- `\frac`, 85, 86, 90, 94, 99-102, 105, 106, 148, 149
- `\frame`, 137
- `\framebox`, 54, 57, 68, 130
- francais-bst**, 113, 121
- `\frenchfigurename`, 75
- `\frenchtablename`, 75
- `\frontmatter`, 31
-
- `\Gamma`, 92, 94, 103, 104
- `\gamma`, 92
- `gather`, 96
- `gather*`, 95
- `\gcd`, 89
- geometry**, 143
- `\geq`, 93, 99, 100
- `\gets`, 94
- graphics**, 69
- graphicx**, 41, 69, 71, 143
-
- `\hat`, 94
- `\height`, 70
- `\hfill`, 57
- `\hom`, 89
- `\href`, 126
- `\HUGE`, 45
- `\Huge`, 45
- `\huge`, 45
- hyperref**, 32, 35, 36, 41, 115, 126, 134, 135, 141, 142, 151, 155
- `\hypersetup`, 134, 135, 151, 152
- `\hyphenation`, 131
-
- `\I`, 14
- icomma**, 20-22, 155
- `\ieme`, 55
- `\iiint`, 86
- `\iint`, 86
- `\in`, 14, 34, 36, 93
- `\include`, 38-40
-
- `\includegraphics`, 41, 42, 70, 71, 77
- `\includeonly`, 39-42
- `\includepdf`, 72, 148
- `\inf`, 89
- `\infty`, 90, 94, 95
- `\input`, 38, 39, 42
- inputenc**, 19, 21
- `\int`, 86, 90, 95, 99, 101
- `\iota`, 92
- `\item`, 46, 47
- `itemize`, 46, 47
- `\itshape`, 44
-
- `\kappa`, 92
- `\ker`, 89
- knitr**, 156
-
- `\label`, 34-36, 76, 96
- `\Lambda`, 92
- `\lambda`, 92, 94, 95
- `\language`, 136
- `\LARGE`, 45
- `\Large`, 44, 45
- `\large`, 8, 45
- `\LaTeX`, 12
- `\ldots`, 87
- `\le`, 9, 14, 56
- `\left`, 98-102, 105, 148, 149
- `\Leftarrow`, 94
- `\leftarrow`, 94
- `\Leftrightarrow`, 94
- `\leftrightharrow`, 94
- `\leq`, 93, 149
- letter, 10, 23
- `\lg`, 89
- `\lim`, 89, 90
- `\liminf`, 89
- `\limsup`, 89
- `\line`, 69

- `\linewidth`, 15, 57, 77, 130, 147, 151
- listings**, 51, 136, 142, 154
- `\listoffigures`, 75
- `\listoftables`, 75
- `\ln`, 89
- `\log`, 89
- `\long`, 12
- `\Longleftarrow`, 94
- `\longleftarrow`, 94
- `\Longleftrightarrow`, 94
- `\longleftrightarrow`, 94
- `\Longrightarrow`, 94
- `\longrightarrow`, 94
- longueur, 14
- lorem**, 28
- `lstlisting`, 136, 152

- `\mainmatter`, 31
- `\makebox`, 54, 68
- `\MakeLowercase`, 128, 150
- `\maketitle`, 28
- `\MakeUppercase`, 150
- `\marginpar`, 126, 150
- `\mathbb`, 92, 124
- `\mathbf`, 103
- `\mathcal`, 92
- `\max`, 89, 98
- `\mbox`, 54, 59, 146
- `\mdseries`, 44
- memoir, viii, ix, 26, 27, 29-31, 33, 41, 43, 45, 50, 51, 63, 65, 68, 74, 78, 123, 125, 130, 143, 147, 166
- `\midrule`, 65, 66, 74, 76, 146, 147
- `\min`, 89
- minipage, 56, 58, 59, 68, 77, 78, 130, 146-148
- `\miniscule`, 45
- `\mu`, 92
- `\multicolumn`, 64, 66, 147

- multiline, 96
- multiline*, 95

- `\nabla`, 94
- natbib**, 107-109, 112-115, 117, 121, 143
- `\ne`, 9, 14, 50
- `\neg`, 94
- `\neq`, 93
- `\newcommand`, 104, 124-126, 128, 150
- `\newenvironment`, 126, 127, 151
- `\newline`, 77, 129, 130
- `\newpage`, 130
- `\newsfloat`, 79, 147
- `\ngeq`, 93
- `\ngtr`, 93
- `\nLeftarrow`, 94
- `\nleftarrow`, 94
- `\nLeftrightarrow`, 94
- `\nleftrightarrow`, 94
- `\nleq`, 93
- `\nless`, 93
- `\No`, 78
- `\nocite`, 115
- NoHyper, 116
- `\nombre`, 21, 78
- `\normalfont`, 43, 44
- `\normalsize`, 45
- `\not`, 14, 93
- `\notag`, 97
- `\notin`, 93
- `\nrightarrow`, 94
- `\nrightrightarrow`, 94
- `\nu`, 92
- numprint**, 21, 22, 78, 143, 146, 154

- `\o`, 18
- `\OE`, 18
- `\oe`, 18
- `\Omega`, 92

- `\omega`, 92
- `\oplus`, 93
- `\otimes`, 93
- package, 139
- `\pagebreak`, 131
- `\pageref`, 34
- paquetage
 - amsmath, 92, 106
 - amsmath, 82, 83, 85–88, 90, 93–97, 100, 102, 104, 155
 - babel, 11, 17, 20–22, 47, 55, 75, 78, 112, 117, 121, 131, 143, 146
 - babel-french, 20, 47
 - biblatex, 108
 - bm, 103, 153
 - booktabs, 62, 63, 65
 - csquotes, 18, 154
 - enumitem, 47, 153
 - fontawesome5, 126, 154
 - fontenc, 20, 21, 142
 - fontspec, 19, 21, 28, 117, 132, 141, 143, 155
 - footmisc, 50, 153
 - francais-bst, 113, 121
 - geometry, 143
 - graphics, 69
 - graphicx, 41, 69, 71, 143
 - hyperref, 32, 35, 36, 41, 115, 126, 134, 135, 141, 142, 151, 155
 - icomma, 20–22, 155
 - inputenc, 19, 21
 - knitr, 156
 - listings, 51, 136, 142, 154
 - lorem, 28
 - natbib, 107–109, 112–115, 117, 121, 143
 - numprint, 21, 22, 78, 143, 146, 154
 - pdfpages, 72, 79, 148, 154
 - tabularx, 63
 - textcomp, 143
 - unicode-math, 21, 92, 103, 105, 155
 - xcolor, 132, 133, 137, 142, 143, 151, 154
- `\par`, 45, 47
- `\parbox`, 55, 56, 58, 145, 146
- `\partial`, 94, 104, 105, 148, 149
- pdfpages**, 72, 79, 148, 154
- `\phantom`, 96, 97, 100, 101
- `\Phi`, 92
- `\phi`, 92
- `\Pi`, 92
- `\pi`, 92, 137, 138
- picture, 67–69
- `\pm`, 93
- pmatrix, 106
- `\Pr`, 89
- `\prime`, 94, 148
- `\Psi`, 92
- `\psi`, 92
- `\put`, 68, 69
- `\qbezier`, 69
- `\quad`, 88, 89
- `\quad`, 88, 89, 95, 99, 105
- quotation, 48
- quote, 13, 14, 48, 127, 151
- `\raisebox`, 55, 150
- `\ref`, 34, 35, 41, 76
- `\reflectbox`, 71, 72, 77
- `\renewcommand`, 124, 125, 128
- `\renewenvironment`, 126
- report, 10, 23–25, 27, 30
- `\resizebox`, 71, 72
- `\rho`, 92
- `\right`, 98–102, 105, 148, 149

- `\Rightarrow`, 94
- `\rightarrow`, 94
- `\rmfamily`, 44
- `\rotatebox`, 71, 72
- `\rule`, 12, 57, 58, 66

- `\scalebox`, 71, 72
- `\scriptsize`, 45
- `\scshape`, 44
- `\sec`, 34, 36, 37, 89
- `\section`, 8, 11, 30, 34, 36, 37, 137, 138
- `\section*`, 11
- `\setmainfont`, 141
- `\setminus`, 93
- `\setmonofont`, 142
- `\setsansfont`, 141
- `\setsecheadstyle`, 43
- `\Sexpr`, 138
- `\sffamily`, 44
- `\Sigma`, 92
- `\sigma`, 92
- `\sin`, 89
- `\sinh`, 89
- `Sinput`, 138
- slides, 10, 23, 24, 139
- `\slshape`, 44
- `\small`, 45
- souignement, 46
- `Soutput`, 138
- split, 96, 97
- `\sqrt`, 86
- `\star`, 93
- `\subcaption`, 79, 147
- `\subset`, 93
- `\subseteq`, 93
- `\sum`, 86, 95, 98, 99, 101, 102
- `\sup`, 89
- `\symbf`, 103, 104

- table, 73, 74, 76, 79, 147, 148
- tabular, 63–66, 74, 76, 146–148
- tabular*, 63, 64
- tabularx, 63, 64, 66, 147, 151
- tabularx**, 63
- `\tan`, 89
- `\tanh`, 89
- `\tau`, 92
- `\text`, 88, 95, 98
- `\textbf`, 44, 103
- `\textcolor`, 133
- textcomp**, 143
- `\textit`, 8, 44, 45
- `\textmd`, 44
- `\textrm`, 44
- `\textsc`, 44, 127, 150
- `\textsf`, 44
- `\textsl`, 44
- `\textsubscript`, 55
- `\textsuperscript`, 16, 55
- `\texttt`, 44
- `\textup`, 44
- `\textwidth`, 15, 42, 56, 63, 64, 66, 147
- `\tfrac`, 85, 106
- `\thanks`, 27, 28
- `\the`, 14, 45
- `\Theta`, 92
- `\theta`, 92, 103–105
- `\thicklines`, 68, 69
- `\thinlines`, 69
- `\tilde`, 94, 149
- `\times`, 93, 105
- `\tiny`, 45
- `\title`, 27, 28
- titlepage, 28
- titlingpage, 28
- `\to`, 90, 94
- `\toprule`, 65, 66, 74, 76, 146, 147, 151

`\ttfamily`, 44

`ulthese`, 121, 141, 143, 153

`unicode-math`, 21, 92, 103, 105, 155

`\unitlength`, 68

`\Uparrow`, 94

`\uparrow`, 94

`\Updownarrow`, 94

`\updownarrow`, 94

`\upshape`, 44

`\Upsilon`, 92

`\upsilon`, 92

`\varepsilon`, 92

`\varphi`, 92

`\varpi`, 92

`\varrho`, 92

`\varsigma`, 92

`\vartheta`, 92

`\vdots`, 87

`\vector`, 69

`\vee`, 93

`\verb`, 47, 50, 137, 138

`\verb*`, 50

`verbatim`, 50, 135

`verbatim*`, 50

`\vert`, 47

`Vmatrix`, 106

`vmatrix`, 106

`\wedge`, 93

`\width`, 42

`xcolor`, 132, 133, 137, 142, 143, 151,
154

`\Xi`, 92

`\xi`, 92

`\zeta`, 92

Ce document a été produit avec le système de mise en page Xe_{La}TeX à partir de la classe memoir. Le texte principal est en Lucida Bright OT 11 points, les mathématiques en Lucida Bright Math OT, le code informatique en Lucida Grande Mono DK et les titres en Fira Sans. Les icônes proviennent de la police Font Awesome.

