Rédaction de thèses
et de mémoires avec LaTeX

\title{LaTeX}

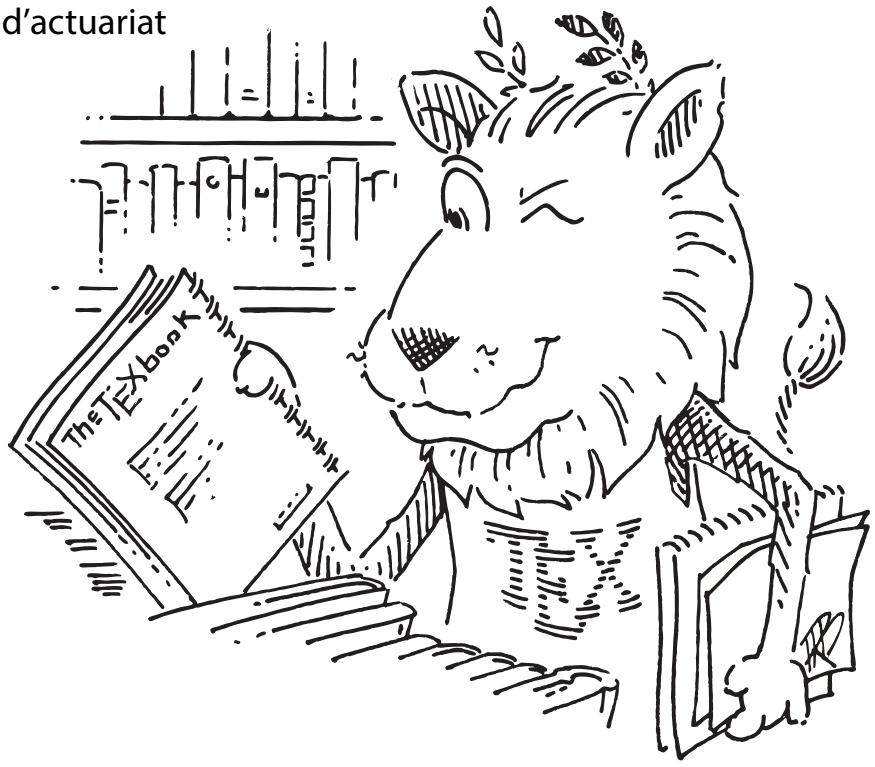
2 | Concepts avancés

UNIVERSITÉ LAVAL

Rédaction de thèses et de mémoires avec LaTeX

2 | Concepts avancés

Vincent Goulet
Professeur titulaire | École d'actuariat



Édition 2016

© 2016 Vincent Goulet, Université Laval

Cette création est mise à disposition selon le contrat Attribution-Partage dans les mêmes conditions 4.0 International de Creative Commons. En vertu de ce contrat, vous êtes libre de :

- **partager** — reproduire, distribuer et communiquer l'œuvre ;
- **remixer** — adapter l'œuvre ;
- utiliser cette œuvre à des fins commerciales.

Selon les conditions suivantes :

Attribution — Vous devez créditer l'œuvre, intégrer un lien vers le contrat et indiquer si des modifications ont été effectuées à l'œuvre. Vous devez indiquer ces informations par tous les moyens possibles, mais vous ne pouvez 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 modifiez, transformez ou créez à partir du matériel composant l'œuvre originale, vous devez diffuser l'œuvre modifiée dans les même conditions, c'est à dire avec le même contrat avec lequel l'œuvre originale a été diffusée.

Notes de cours et exercices développés par Vincent Goulet avec la contribution financière de la Bibliothèque de l'Université Laval.

ISBN 978-2-9811416-5-1
Dépôt légal – Bibliothèque et Archives nationales du Québec, 2016
Dépôt légal – Bibliothèque et Archives Canada, 2016

Code source

Le code source de ce document est conservé dans un dépôt Subversion public.

Crédits

Couverture réalisée par Marie-Ève Guérard et Hélène Coulouarn. Lion de CTAN réalisé par Duane Bibby.

Introduction

Ce document constitue la seconde partie d'une formation sur la rédaction de thèses et de mémoires avec LaTeX développée pour la Bibliothèque de l'Université Laval. La première partie de la formation se déroulant en classe, la documentation qui l'accompagne consiste en une série de diapositives (Goulet, 2015).

Nous reprenons ici la formation une fois présentés les concepts de base de LaTeX 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. Avec cette seconde partie, une personne devrait être en mesure de composer des documents relativement complexes comportant des tableaux, des figures, des équations mathématiques élaborées, une bibliographie, etc.

Le présent ouvrage n'a aucune prétention d'exhaustivité. La consultation de documentation additionnelle peut s'avérer nécessaire pour réaliser des mises en page plus élaborées. À cet égard, nous recommandons 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, 2013), sur laquelle se base la classe ulthese pour les thèses et mémoires de l'Université Laval, constitue une autre référence de choix. Nous recommandons également :

- *LaTeX* dans Wikilivre pour de la documentation en ligne, en français et libre ;
- le très actif forum de discussion TeX-LaTeX Stack Exchange (avant de penser y poser une question, vérifier que la réponse ne se trouve pas déjà dans le forum… ce qui risque fort d'être le cas) ;
- la très complète *foire aux questions* (en anglais) du groupe des utilisateurs de LaTeX du Royaume-Uni.

Mode d'emploi

Hyperliens vers la documentation

À plusieurs endroits dans le document nous renvoyons le lecteur vers la documentation d'un paquetage ou d'une classe, par exemple vers la documentation de la classe ulthese. Le format du renvoi est toujours tel qu'illustré ici :

`ulthese`

- un hyperlien mène vers la version en ligne de la documentation dans le site TeXdoc Online ;
- on trouve dans la marge le nom du fichier correspondant (sans l'extension `.pdf`) sur un système doté de TeX Live.

Sur la plupart des systèmes, il est possible de consulter hors ligne le fichier de documentation ⟨*fichier*⟩`.pdf` en entrant à une invite de commande

```
texdoc ⟨fichier⟩
```

Fichiers d'accompagnement

Ce document devrait être accompagné de fichiers nécessaires pour compléter certains exercices figurant à la fin des chapitres, ainsi que d'un fichier `exercice_gabarit.tex` pouvant servir de gabarit pour composer les solutions des autres exercices.

Si vous n'avez pas obtenu ces fichiers avec le document, vous pouvez les récupérer dans le site *Comprehensive TeX Archive Network* (CTAN).

Accéder aux fichiers dans CTAN

Table des matières

Introduction v

Mode d'emploi vii
Hyperliens vers la documentation vii
Fichiers d'accompagnement vii

1 Document contenu dans plusieurs fichiers 1
1.1 Insertion du contenu d'un autre fichier 1
1.2 Insertion de parties d'un document 2
1.3 Exercices 4

2 Boîtes 7
2.1 Longueurs 7
2.2 Boîtes horizontales 8
2.3 Boîtes verticales 10
2.4 Boîtes de réglure 12
2.5 Exercices 12

3 Tableaux et figures 15
3.1 De la conception de beaux tableaux 15
3.2 Tableaux 17
3.3 Figures et graphiques 21
3.4 Éléments flottants 27
3.5 Exercices 31

4 Mathématiques 35
4.1 Rappel des principes de base 35
4.2 Un paquetage incontournable 37
4.3 Principaux éléments du mode mathématique 38
4.4 Symboles mathématiques 45

4.5 Équations sur plusieurs lignes et numérotation 50
4.6 Délimiteurs de taille variable 53
4.7 Caractères gras en mathématiques 57
4.8 Exercices 60

5 Bibliographie et citations 63
5.1 Quel système utiliser ? 63
5.2 Processus de création d'une bibliographie 65
5.3 Création d'une base de données 66
5.4 Style des citations et de la bibliographie 69
5.5 Insertion de références dans le texte 71
5.6 Composition de la bibliographie 72
5.7 Exercices 77

6 Commandes et environnements définis par l'usager 79
6.1 Nouvelles commandes 80
6.2 Nouveaux environnements 82
6.3 Exercices 84

7 Trucs et astuces divers 87
7.1 Polices de caractères : au-delà de Computer Modern 87
7.2 Couleurs 88
7.3 Hyperliens et métadonnées de documents PDF 90
7.4 Présentation de code informatique 91
7.5 Production de rapports avec l'analyse intégrée 93
7.6 Diapositives 96
7.7 Gestion des versions et travail collaboratif 96
7.8 Exercices 97

A Solutions des exercices 101
Chapitre 2 101
Chapitre 3 102
Chapitre 4 104
Chapitre 5 105
Chapitre 6 106
Chapitre 7 107

Bibliographie 109

Index 113

1 Document contenu dans plusieurs fichiers

Un document LaTeX comporte toujours un préambule suivi du corps du texte. Lorsque ceux-ci sont 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 son 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 divers éléments 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.

1.1 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`. On laisse donc tomber l'extension `.tex`, qui est implicite. Le contenu du fichier est inséré tel quel dans le document, comme s'il avait été tapé dans le fichier qui contient l'appel à `\input`.

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

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.

1.2 Insertion de parties d'un document

ulthese

Extrait de la documentation de la classe ulthese :

> « Il est recommandé de segmenter tout document d'une certaine ampleur dans des fichiers `.tex` distincts pour chaque partie — habituellement un fichier par chapitre. Le document complet est composé à l'aide d'un fichier maître qui contient le préambule LaTeX et un ensemble de commandes `\include` pour réunir les parties dans un tout. »

Comme `\input`, la commande `\include` insère le contenu d'un autre fichier dans un document LaTeX. Son effet est cependant différent et c'est son utilisation qui permet d'accélérer la compilation d'un long document.

L'insertion d'un fichier avec `\include` débute toujours une nouvelle page. On utilisera donc `\include` 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`.

La syntaxe de la commande `\include` est

```
\include{⟨fichier⟩}
```

où le nom du fichier à insérer est ⟨*fichier*⟩`.tex`. Ici aussi on laisse tomber l'extension `.tex` qui est implicite.

La structure type d'un fichier maître est la suivante :

```
\documentclass{ulthese}
  [...]

\begin{document}

\frontmatter

\include{introduction}
\tableofcontents*

\mainmatter
```

Conseil du TEXpert

Utilisez des noms des fichiers qui permettent de facilement identifier leur contenu. Par exemple, un nom comme `rappels.tex` est plus parlant et résiste mieux aux changements à l'ordre des chapitres que `chapitre1.tex`.

```
\include{historique}            % premier chapitre
\include{modele}                % deuxième chapitre
[...]

\end{document}
```

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 références. Cela permet, par exemple, de compiler le texte d'un seul chapitre — plutôt que le document entier — et néanmoins obtenir une image représentative du chapitre. Procéder ainsi accélère significativement la compilation des documents longs ou complexes.

La commande `\includeonly`, que l'on utilise 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 dans les fichiers ⟨*liste_fichiers*⟩ suivra celle établie lors de la compilation précédente. Si l'édition des fichiers de ⟨*liste_fichiers*⟩ 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 1.1. 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 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 renumérotées à partir de 47.

□

L'exercice 1.1 illustre mieux le cycle typique d'utilisation des commandes `\include` et `\includeonly`.

1.3 Exercices

1.1 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 presentation.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 `exercice_include.tex`, puis le compiler deux à trois fois jusqu'à ce que toutes les références internes 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{presentation}
```

Vers la fin du fichier `presentation.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 maître la commande `\includeonly`, puis compiler de nouveau le document quelques fois.

1.2 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`.

2 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, on aura souvent recours aux tableaux 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*). Les commandes et environnements présentés ci-dessous permettent simplement de créer d'autres boîtes dont le contrôle des dimensions et du contenu est laissé à l'usager.

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

2.1 Longueurs

Plusieurs des commandes de ce chapitre requièrent de spécifier une largeur ou une hauteur. Dans la terminologie de LaTeX, 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

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

les principales unités de mesure utilisées par LaTeX et le symbole correspondant.

Il existe un certain nombre de longueurs prédéfinies. Les plus utiles sont

```
\linewidth
\textwidth
```

La première contient la largeur de la ligne de texte courante et la seconde, la largeur de la page courante. Dans du texte normal, les deux mesures sont habituellement égales.

2.2 Boîtes horizontales

Le plus simple concept de boîte dans LaTeX est celui de boîte « horizontale », c'est-à-dire dont le contenu est disposé latéralement de gauche à droite [1]. Le 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⟩}
```

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

Conseil du TEXpert

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 2.2.

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 .

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. Ainsi

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

produit $^{\text{au-dessus}}$ de la ligne de base et

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

produit $_{\text{au-dessous}}$ de la ligne de base.

Attention, toutefois, de ne pas utiliser `\raisebox` pour placer du texte en exposant ou en indice. Selon la nature du texte, employer plutôt les commandes `\textsuperscript` et `\textsubscript`, les commandes de la famille `\ieme` de **babel** (section 1.1 de la documentation) ou, pour des sym- frenchb

boles mathématiques, les commandes d'exposant et d'indice spécifiques au mode mathématique (section 4.3.1).

2.3 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 paragraphes [2].

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 comme il faut bien définir la largeur de la page pour le texte normal.

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 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}[⟨pos⟩]{⟨largeur⟩}
  ⟨texte⟩
\end{minipage}
```

La signification des arguments ⟨*largeur*⟩ et ⟨*pos*⟩ est la même que pour la commande `parbox`.

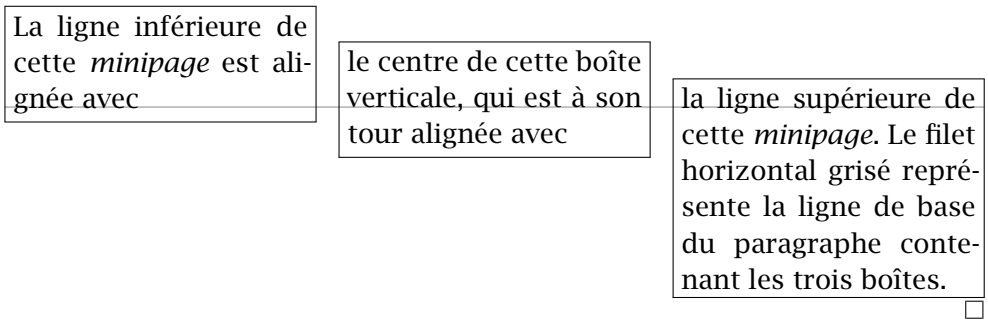
L'environnement `minipage` est fréquemment utilisé pour disposer des éléments de contenu de manière spécifique sur la page, notamment des tableaux ou des figures côte à côte ou en grille (voir l'exemple 3.7 à la page 31).

Exemple 2.1. Le code ci-dessous

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

```
\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}
```

produit :



La commande `\hfill` utilisée entre les boîtes dans l'exemple ci-dessus indique à LaTeX d'insérer de l'espace blanc 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

alors que

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

produit

2.4 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 boîte[3] dans LaTeX.

La commande

```
\rule[⟨déplacement⟩]{⟨largeur⟩}{⟨hauteur⟩}
```

crée une réglure de dimensions ⟨*largeur*⟩ × ⟨*hauteur*⟩. Par défaut, la réglure s'appuie sur la ligne de base. Le résultat de

```
\rule{2cm}{6pt}
```

est donc une ligne pleine de 2 cm de long et de 6 points d'épais : .

L'argument optionnel ⟨*déplacement*⟩ permet de déplacer verticalement la réglure au-dessus ou au-dessous de la ligne de base selon que la longueur ⟨*déplacement*⟩ est positive ou négative. Avec les deux commandes

```
\rule[3pt]{2cm}{6pt}
\rule[-3pt]{2cm}{6pt}
```

on crée 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 parait en insérant dans celle-ci une réglure de largeur nulle. Par exemple, la distance entre la présente ligne et les autres du paragraphe est plus grande que la normale parce que nous y avons 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 3.2.

2.5 Exercices

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

2.1 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 :

3. *Rule box*, en anglais

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

2.2 Réaliser l'agencement de boîtes verticales suivant :

Deux boîtes verticales de hauteurs différentes placées côte à côte

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).

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) :

Les deux boîtes sont correctement alignées l'une par rapport à l'autre

mais l'alignement avec la ligne de base est incorrect.

La raison : pour TEX, 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.

2.3 Réaliser l'agencement de boîtes verticales ci-dessous. (La taille de la police de caractères 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.

3 Tableaux et figures

Les tableaux et graphiques ne sont pas les éléments de texte les plus simples et rapides à créer avec LaTeX. Les traitements de texte brillent, ici, 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 ! Peut-être avez-vous déjà eu de la difficulté à contrôler les bordures d'un tableau, la hauteur des lignes ou la largeur des colonnes dans un traitement de texte ? Alors vous comprenez que l'exercice de composition d'un tableau 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.

3.1 De la conception de beaux tableaux

On utilise les tableaux pour 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 filets [1] horizontaux et verticaux.

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

Comparer les deux tableaux ci-dessous. Le premier est mis en forme selon une approche classique supportée depuis toujours par LaTeX : filets

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*.

doubles en entête et en pied de tableau, filets simples entre chaque ligne et entre les colonnes.

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, 2005) 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 ?

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 :

booktabs

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.

Nous recommandons évidemment de suivre ces règles et c'est pourquoi la présente documentation ainsi que les fichiers d'exemples font usage des commandes de **booktabs**.

Les fonctionnalités de **booktabs** sont intégrées à la classe memoir et, par conséquent, à ulthese. Il n'est donc pas nécessaire de charger le paquetage avec ces deux classes.

3.2 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}{⟨format⟩}           lignes \end{tabular}
\begin{tabular*}{⟨largeur⟩}{⟨format⟩} lignes \end{tabular*}
\begin{tabularx}{⟨largeur⟩}{⟨format⟩} lignes \end{tabularx}
```

La signification des arguments [2] est la suivante. Nous ne traitons ici que les options les plus souvent employées. Pour une liste plus exhaustive, consulter la documentation de la classe memoir (chapitre 11) ou Wikilivres (2015, section Tableaux). memman

⟨*largeur*⟩ Largeur hors tout d'un tableau avec les environnements `tabular*` et `tabularx`. Autrement, avec 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. Celle-ci est accessible avec la commande `\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 :

2. Nous avons omis un argument optionnel à peu près jamais utilisé servant à spécifier l'alignement vertical du tableau par rapport à la ligne de base externe.

```
\begin{tabular*}{\textwidth}{⟨format⟩}
```

```
\begin{tabularx}{0.8\textwidth}{⟨format⟩}
```

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 seconde est aligné à droite ; celui de la troisième est en texte libre dans une cellule de 5 cm de largeur.

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 3.1 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.

Les lignes de contenu peuvent également contenir certaines commandes spéciales pour contrôler la mise en forme. La commande ci-dessous permet de fusionner des cellules :

`\multicolumn{`⟨*num*⟩`}{`⟨*fmt*⟩`}{`⟨*texte*⟩`}` fusionne les ⟨*num*⟩ cellules suivantes en une seule de format ⟨*fmt*⟩ et contenant le texte ⟨*texte*⟩.

Cette commande ne peut apparaître qu'au début d'une ligne ou après un symbole de changement de colonne &.

La commande est souvent utilisée avec une valeur de ⟨*num*⟩ 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.

Les commandes suivantes[3] servent à insérer des filets horizontaux dans un tableau :

`\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{`⟨*m*-*n*⟩`}` insère un filet horizontal comme `\midrule` de la gauche de la colonne ⟨*m*⟩ à la droite de la colonne ⟨*n*⟩ ;

`\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 3.1. On crée d'abord un tableau d'une largeur ajustée automatiquement au contenu. Remarquer comment les lignes de contenu sont définies. La première colonne est alignée à gauche et toutes les autres, à droite.

```
\begin{tabular}{lrrr}
  \toprule
  Produit & Quantité & Prix unitaire (\$) & Prix (\$) \\
  \midrule
```

3. Ce sont les commandes de **booktabs** et memoir auxquelles nous faisions référence à la section 3.1.

```
  Vis à bois    & 2 & 9,90 & 19,80 \\
  Clous vrillés & 5 & 4,35 & 21,75 \\
  \midrule
  TOTAL         & 7 &      & 41,55 \\
  \bottomrule
\end{tabular}
```

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

Avec quelques modifications, le tableau occupera toute la largeur de la page, la largeur de la première colonne étant ajustée pour combler l'espace nécessaire. De plus, on modifie l'entête de la première colonne avec la commande `\multicolumn` afin de centrer le titre. Enfin, on augmente la hauteur de l'entête à l'aide d'une réglure invisible (section 2.4).

```
\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}
```

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

□

3.3 Figures et graphiques

Il est possible de tracer des figures simples directement avec LaTeX. Par « simple » on entend : 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, assez pratique puisque le code source d'une figure se trouve alors dans le même format que le reste du document.

Pour la création de figures et de graphiques plus complexes, on aura 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 document. Le reste de cette section ne présente que les principales fonctionnalités. Le lecteur qui souhaite en savoir plus pourra se référer aux sources de documentation habituelles figurant à la bibliographie.

3.3.1 Figures LaTeX

L'environnement `picture` permet de tracer des figures simples dans LaTeX comme des diagrammes à base de texte, des flux logiques ou des organigrammes. Quelques logiciels spécialisés de création de graphiques sont 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 de caractère du texte de la figure sera le même que celle du document.

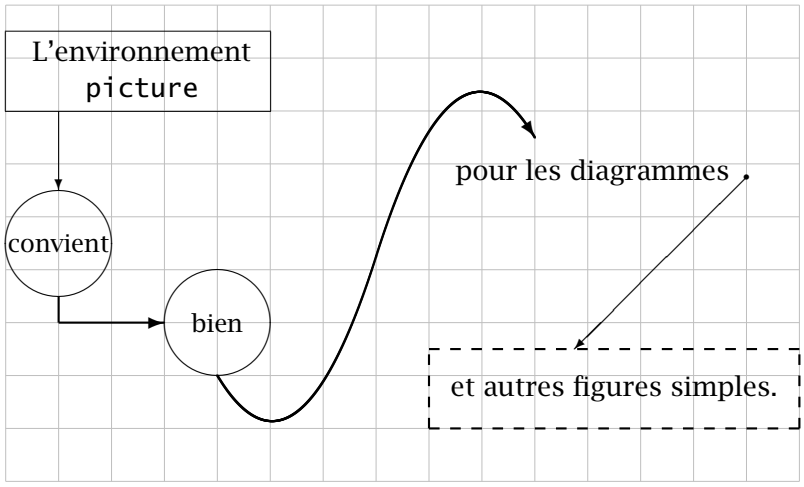
Pour tracer une figure avec l'environnement `picture`, on crée d'abord une grille (invisible) d'une dimension quelconque dans l'unité de mesure de son choix (autrement dit : les lignes de la grille peuvent être distantes aussi bien de `1pt` que de `1cm`). Ensuite, on dispose des éléments sur la grille en donnant les coordonnées du point d'ancrage et, le cas échéant, les dimensions de l'élément, la distance à parcourir ou quelqu'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 permettre de comprendre les principes de base de la création de figures. Autrement, l'annexe D de la documentation de la classe memoir fournit une

memman

bonne introduction à `picture`.

(Nous avons 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 boîtes
  %%%
  %% 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 boîte
        \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 boîtes
  %%%
  %% 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 Ti*k*Z/PGF (Tantau, 2014). Ce dernier gagne beaucoup en popularité depuis quelques années.

3.3.2 Importation d'images

Il est aujourd'hui simple d'importer des images de source externes dans un document LATEX en utilisant l'un ou l'autre des paquetages **graphics** ou **graphicx** (Carlisle et The LATEX3 Project, 2014) en combinaison avec un moteur TEX moderne tel que pdfLATEX ou XELATEX. Les fonctionnalités des deux paquetages sont les mêmes, seules les syntaxes des commandes diffèrent. Nous présenterons les commandes de **graphicx**, plus modernes et conviviales.

La commande de base pour importer des images dans un document LATEX 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 pdfLaTeX et XeLaTeX, 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.

grfguide

Exemple 3.2. Le fichier `ul_p.pdf` contenant le logo de l'Université Laval en couleur et en format vectoriel est distribué avec la présente documentation ainsi qu'avec la classe ulthese. 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` :

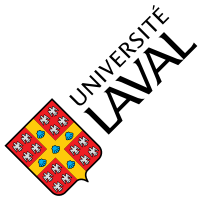
```
%% réduction à 40 % de taille réelle
\includegraphics[scale=0.4]{ul_p}
```

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

(Il est préférable d'utiliser une seule de `width` ou `height`. Autrement, ajouter 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 % et rotation à 45 degrés
\includegraphics[angle=45,scale=0.25]{ul_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 (`bp` ; 1 bp = 1/72 pouce) 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]{ul_p}
```

□

La commande `\includegraphics` permet d'appliquer certaines transformations aux images importées, mais celles-ci 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{ul_p}}
```

et

```
\includegraphics[angle=45]{ul_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 ⟨*dim-h*⟩ ou ⟨*dim-v*⟩ et `!` pour l'autre valeur pour éviter de déformer ⟨*texte*⟩.

Exemple 3.3. Voici des exemples d'utilisation des commandes `\rotatebox`, `\scalebox`, `\resizebox` et `\reflectbox` avec du texte :

```
\rotatebox{135}{texte}
```
texte

```
\scalebox{1.5}{texte}
```
texte

```
\scalebox{1.5}[0.75]{texte}
```
texte

```
\resizebox{3cm}{!}{texte}
```
texte

```
\reflectbox{texte}
```
texte

□

3.3.3 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 pdfLaTeX ou XeLaTeX, le très pratique paquetage **pdfpages** (Matthias, 2015) fournit la commande

```
\includepdf[⟨options⟩]{⟨fichier⟩}
```

pdfpages

Les ⟨*options*⟩ sont trop nombreuses pour les présenter ici ; consulter la documentation.

Exemple 3.4. Les couvertures avant et arrière du présent document ont été réalisées dans un logiciel spécialisé de création graphique, sauvegardées dans un fichier `couvertures.pdf`, puis mises en place dans le document avec

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

aux endroits appropriés. □

3.4 Éléments flottants

Dans la terminologie de LaTeX, un élément flottant [4] 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 voudrait-on laisser LaTeX décider où un élément de contenu devrait se retrouver dans notre document ? D'abord et avant tout pour les tableaux et les figures.

En effet, les tableaux et les figures sont des éléments de contenu qui occupent souvent beaucoup d'espace vertical dans la page. S'il ne reste plus assez de place pour y afficher un tel élément, LaTeX devra le déplacer au début de la page suivante et cela risque de produire une page inesthétique car insuffisamment remplie [5]. Les traitements de texte génèrent sans rechigner des pages à demi remplies dans de telles situations.

En définissant un élément comme flottant, on 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.

On crée des éléments flottants avec les environnements `table` et `figure` :

```
\begin{table}[⟨pos⟩]  ⟨tableau⟩ \end{table}
\begin{figure}[⟨pos⟩] ⟨figure⟩ \end{figure}
```

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

L'argument optionnel ⟨*pos*⟩ 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 ⟨*pos*⟩ 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 ⟨*pos*⟩ ;

`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.

memman

La valeur par défaut de l'argument ⟨*pos*⟩ est `tbp`. La section 10.4 de la documentation de memoir explique plus en détail la signification des valeurs ci-dessus. Le lecteur qui voudrait vraiment *tout* savoir sur la disposition des éléments flottants pourra consulter Mittelbach (2014).

Exemple 3.5. On reprend le tableau de l'exemple 3.1, 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}
```

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

Conseil du TEXpert

Les anciennes version du style français de **babel** utilisaient les étiquettes plus neutres « TAB. » et « FIG. » dans les légendes des tableaux et figures. Pour utiliser ces versions plutôt que les versions par défaut — comme dans le présent document — ajoutez dans le préambule les commandes suivantes :

```
\def\frenchtablename{{\scshape Tab.}}
\def\frenchfigurename{{\scshape Fig.}}
```

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 figures [6] 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 apparait 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.

Pour faire référence à un tableau ou à une figure dans le texte, il faut utiliser le système de renvois automatiques de LaTeX (Goulet, 2015, section 4). On attribue une étiquette à l'élément flottant en plaçant la commande `\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.

6. Obtenues respectivement avec `\listoftables` et `\listoffigures` (Goulet, 2015, section 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. 3.1 – Code source pour créer le tableau 3.1

TAB. 3.1 – Tableau correspondant au code de la figure 3.1

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

L'exemple suivant présente finalement la recette complète pour composer un tableau et une figure dans LaTeX, légende et renvoi inclus.

Exemple 3.6. Attention, exemple récursif : son texte constitue lui-même un exemple. Le code source de la figure 3.1 crée le tableau 3.1. □

Les environnements `table` et `figure` créent des éléments flottants qui, par ailleurs, sont des boîtes verticales standards (section 2.3). 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 environnements `minipage` (section 2.3) se revèlent alors particulièrement utiles pour disposer les éléments de contenu dans la boîte.

À l'exercice 3.3, on montre comment ajouter des sous-légendes pour chacun des éléments. La section 10.9 de la documentation de memoir comporte de nombreux détails additionnels sur les sous-légendes.

memman

Exemple 3.7. Le code ci-dessous démontre comment disposer quatre images sous forme de grille 2×2 dans une même figure flottante à l'aide de boîtes verticales créées avec l'environnement `minipage`. On pourrait faire de même avec des tableaux.

Dans la figure 3.2 correspondant au code, nous avons indiqué en grisé les limites des boîtes verticales.

```
\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}
```

□

3.5 Exercices

3.1 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**.

FIG. 3.2 – Exemple de disposition de plusieurs graphiques dans une même figure flottante

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

3.2 Apporter au tableau de l'exercice précédent les modifications suivantes :

i) centrer le titre de la deuxième colonne ;

ii) ajuster automatiquement la largeur du tableau au bloc de texte sur la page avec un environnement `tabularx`.

3.3 L'exemple 3.7 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 les classes ulthese et memoir, la production de sous-légendes requiert d'abord de déclarer, dans le préambule du document, son inten-

tion 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{⟨texte⟩}
```

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.

3.4 Insérer, disons, la page couverture du présent document dans un document de votre cru à l'aide des fonctionnalités du paquetage **pdfpages** décrites à la section 3.3.3.

3.5 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.

4 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.

4.1 Rappel des principes de base

Par souci d'exhaustivité, nous revenons d'abord sur les quelques principes de base présentés dans la première partie de cette formation (Goulet, 2015, section 7).

La mise en forme d'équations mathématiques requiert d'indiquer à l'ordinateur, dans un langage spécial, le contenu des dites équations 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 manière 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
$$(a + b)^2 = a^2 + b^2, \tag{4.1}$$
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 avoir recours 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.

4.2 Un paquetage incontournable

Le paquetage **amsmath** (American Mathematical Society, 2002) produit par la prestigieuse *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 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.

Nous décrivons certaines de ces fonctionnalités dans la suite, mais l'utilisateur le moindrement avancé devrait impérativement consulter la documentation complète du paquetage.

amsldoc

4.3 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 section 4.4 présente une partie de l'éventail de symboles mathématiques offerts par LaTeX.

4.3.1 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 $^{\text{exposant}}$ d'un autre avec la commande `^` et en $_{\text{indice}}$ 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. En fait, si TeX rencontre l'une de ces commandes en mode texte, il tentera automatiquement de passer au mode mathématique après avoir émis l'avertissement

```
! Missing $ inserted.
```

Il est assez rare que le résultat soit celui souhaité.

4.3.2 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. Ainsi, on utilise `$(n + 1)/2$` pour obtenir $(n + 1)/2$.

De manière plus générale, la commande

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

dispose ⟨*numérateur*⟩ au-dessus de ⟨*dénominateur*⟩, séparé 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

Conseil du TEXpert

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

fraction au fil du texte (*text*) dans le cas de `\tfrac`. Consulter l'exemple 4.5 à la page 58 pour visualiser l'effet de la commande `\dfrac`.

4.3.3 Racines

La commande

```
\sqrt[⟨n⟩]{⟨radicande⟩}
```

construit un symbole de radical autour de ⟨*radicande*⟩, par défaut la racine carrée. Si l'argument optionnel ⟨*n*⟩ 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}}` $\sqrt{\frac{x+y}{x^2-y^2}}$

4.3.4 Sommes et intégrales

Les sommes et 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 $\sum$ 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 $\iiint$).

On entre les éventuelles limites inférieures et supérieures comme des indices et des exposants.

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

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

TAB. 4.1 – Points de suspension

commande	type de points	exemple
`\dots`	sélection automatique	
`\ldots`	points à la ligne de base	$x_1, \ldots, x_n$
`\cdots`	points centrés	$x_1 + \cdots + x_n$
`\vdots`	points verticaux	x_1 ⋮ x_n
`\ddots`	points diagonaux	x_1 ⋱ x_n

```
\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 4.2 de la page 43.

4.3.5 Points de suspension

Les formules mathématiques comportent fréquemment des points de suspension dans des suites de variables ou d'opérations. On recommande d'éviter de les entrer comme trois points finaux consécutifs, car l'espacement entre les points sera trop petit et le résultat, jugé disgracieux[1] d'un point de vue typographique : ...

Le tableau 4.1 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, nous recommandons d'utiliser principalement cette commande pour insérer des points de suspension en mode mathématique.

1. Le résultat exact dépend de la police de caractère utilisée.

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

```
$x_1, \ldots, x_n$
```
$x_1, \ldots, x_n$

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

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

Le paquetage définit également les commandes sémantiques

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

4.3.6 Texte et espaces

On l'a vu, 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
```

TAB. 4.2 – Espaces dans le mode mathématique

commande	longueur	exemple
	pas d'espace	⊣⊢
\,	3/18 de quad	⊣ ⊢
\:	4/18 de quad	⊣ ⊢
\;	5/18 de quad	⊣ ⊢
\!	−3/18 de quad	⊣⊢
\quad	1 em	⊣ ⊢
\qquad	2 em	⊣ ⊢

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 longueur.[2]

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

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

Le tableau 4.2 répertorie et compare les différentes commandes qui permettent d'insérer des espaces plus ou moins fines entre des éléments dans le mode mathématique.

4.3.7 Fonctions et opérateurs

Les règles de typographie des équations mathématiques veulent 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.

On sait que l'on ne peut entrer le nom d'une fonction tel quel en mode mathématique, car LaTeX interprétera la suite de lettres comme un produit de variables :

```
$2 sin 30$
```

$2sin30$

```
$3 log 2$
```

$3log2$

Or, utiliser à répétition la commande \text pour entrer des noms de fonction se révélerait peu pratique à l'usage.

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

LaTeX définit donc des commandes pour un grand nombre de fonctions et d'opérateurs mathématiques standards :

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

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.

```
% au fil du texte
\lim_{x \to 0} x = 0
```
$\lim_{x \to 0} x = 0$

```
% hors paragraphe
\lim_{x \to 0} x = 0
```
$$\lim_{x \to 0} x = 0$$

Lorsque des usages particuliers requièrent de nouveaux noms de fonctions, la commande `\DeclareMathOperator` de **amsmath** permet de les définir ; consulter la documentation du paquetage (section 5.1) pour les détails.

amsldoc

Exemple 4.1. 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.

On présente ci-dessous, pièce par pièce, le code LaTeX pour créer l'équation suivante :

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

```
\begin{equation*}
\int_x^\infty
(y - x) f_{X|X > x}(y)\, dy =
```
équation hors paragraphe

$$\int_x^\infty (y-x) f_{X|X>x}(y)\,dy =$$

```
\frac{1}{1 - F_X(x)}
```
$$\frac{1}{1 - F_X(x)}$$

```
\int_x^\infty (y - x) f_X(y)\, dy
```
$$\int_x^\infty (y - x) f_X(y)\, dy$$

```
\end{equation*}
```
fin de l'environnement

□

4.4 Symboles mathématiques

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

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

Ceux-ci sont utilisables directement dans les équations. Les accolades `{ }` étant des symboles réservés par LaTeX, il faut les entrer avec `\{` et `\}`, comme dans du texte normal.

Pour représenter les innombrables autres symboles mathématiques, on aura recours à des commandes qui débutent, comme d'habitude, par le symbole `\` 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. Nous nous contenterons d'un avant goût des principales catégories.

L'ouvrage de référence pour connaître les symboles disponibles dans LaTeX est la bien nommée Comprehensive LaTeX Symbol List (Pakin, 2009). La liste comprend près de 6 000 symboles répartis sur plus de 160 pages ! On y trouve de tout, des symboles mathématiques aux pictogrammes de météo ou d'échecs, en passant par… des figurines des Simpsons.

comprehensive

4.4.1 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 4.3 et 4.4 présentent l'ensemble des lettres grecques disponibles dans LaTeX.

Les moteurs XeTeX et LuaTeX supportent nativement le code source en format Unicode UTF-8 (Unicode Consortium, 2007). Ce standard contient des définitions pour plusieurs symboles mathématiques (Wikipedia, 2015). 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. Nous ne saurions toutefois recommander cette pratique qui rend les fichiers source moins compatibles d'un système à un autre.

TAB. 4.3 – Lettres grecques minuscules

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

TAB. 4.4 – Lettres grecques majuscules

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

4.4.2 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.

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

```
\mathcal{ABC\, xyz}
```

$\mathcal{ABC}\,xyz$

La commande `\mathbb` fournie, entre autres, par les paquetages **amsfonts** et **unicode-math** génère des versions majuscule ajourée (*blackboard bold*)

Conseil du TEXpert

Certaines polices de caractères 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 : $\mathcal{ABC}\ xyz$. Consultez éventuellement la documentation de la police pour les détails.

des lettres de l'alphabet. Elles sont principalement utilisée pour représenter les ensembles de nombres.

```
\mathbb{NZRC}
```

$\mathbb{NZRC}$

Le tableau 213 de la Comprehensive LaTeX Symbol List présente plusieurs autres alphabets spéciaux disponibles en mode mathématique.

4.4.3 Opérateurs binaires et relations

Les opérateurs binaires combinent deux quantités pour en former une troisième ; pensons simplement aux opérateurs d'addition + et de soustraction − 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 4.5 présente une sélection d'opérateurs binaires et le tableau 4.6 une sélection de relations.

La Comprehensive LaTeX 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 4.5 et 4.6 de la page 49 ne présentent que les principaux éléments à titre indicatif.

Certaines relations existent directement en version opposée, ou négative (comme ≠ ou ∉) soit dans LaTeX 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 /.

4.4.4 Flèches

Les flèches de différents types sont souvent utilisées en notation mathématique, notamment dans les limites ou pour les expressions logiques. Le tableau 4.7 en présente une sélection.

On retrouve les flèches utilisables en notation mathématique dans les tableaux 102 à 119 de la Comprehensive LaTeX Symbol List. Le document contient divers autres types de flèches, mais celles-ci ne sont généralement pas appropriées pour les mathématiques (pensons à $\rightsquigarrow$ ou $\rightarrow$).

Le paquetage **amsmath** fournit plusieurs flèches additionnelles ainsi que la négation des plus communes. Ces dernières sont d'ailleurs présentées dans le tableau 4.7.

4.4.5 Accents et autres symboles utiles

Le tableau 4.8 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, consulter le tableau 164 de la Comprehensive LaTeX Symbol List. Les versions extensibles de certains accents se trouvent au tableau 169. Quant aux symboles mathématiques divers, on en trouve de toutes les sortes dans les tableaux 201–212.

Exemple 4.2. 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)}{                     \frac{\Gamma(\alpha)}{\lambda^\alpha} =
  \lambda^\alpha} =
\sum_{j = 0}^\infty \int_j^{j + 1}         \sum_{j=0}^{\infty} \int_j^{j+1}
x^{\alpha - 1} e^{-\lambda x}\, dx         x^{\alpha-1} e^{-\lambda x}\, dx
, \quad \alpha > 0 \text{ et }             , \quad \alpha > 0 \text{ et } \lambda > 0.
\lambda > 0.
\end{equation*}                            fin de l'environnement
```

□

TAB. 4.5 – Quelques opérateurs binaires

$\times$	\times	$\div$	\div	$\pm$	\pm	$\cdot$	\cdot
$\cup$	\cup	$\cap$	\cap	$\setminus$	\setminus	$\circ$	\circ
$\wedge$	\wedge	$\vee$	\vee	$\oplus$	\oplus	$\otimes$	\otimes
$*$	\ast	$\star$	\star	$\boxplus$	\boxplus†	$\boxtimes$	\boxtimes†

† requiert **amsmath**

TAB. 4.6 – Quelques relations et leur négation

$\leq$	\leq	$\geq$	\geq	$\neq$	\neq	$\equiv$	\equiv
$\subset$	\subset	$\subseteq$	\subseteq	$\in$	\in	$\notin$	\notin
$\not<$	\nless†	$\not>$	\ngtr†	$\not\leq$	\nleq†	$\not\geq$	\ngeq†

† requiert **amsmath**

TAB. 4.7 – Quelques flèches et leur négation

$\leftarrow$	\leftarrow \gets	$\longleftarrow$	\longleftarrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow
$\rightarrow$	\rightarrow \to	$\longrightarrow$	\longrightarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow
$\uparrow$	\uparrow	$\downarrow$	\downarrow
$\Uparrow$	\Uparrow	$\Downarrow$	\Downarrow
$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\nleftarrow$	\nleftarrow†	$\nleftrightarrow$	\nleftrightarrow†
$\nrightarrow$	\nrightarrow†	$\not\Leftarrow$	\nLeftarrow†
$\not\Leftrightarrow$	\nLeftrightarrow†	$\not\Rightarrow$	\nRightarrow†

† requiert **amsmath**

TAB. 4.8 – Accents et symboles mathématiques divers

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

4.5 Équations sur plusieurs lignes et numérotation

Dans ce qui précède, nous n'avons 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

$$y = 2x + 4 \tag{4.2}$$
$$y = 6x - 1 \tag{4.3}$$

ou des suites d'équations comme

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

Nous recommandons 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 4.9 — 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`.
- Dans les environnements `align` et `split`, les équations successives sont alignées sur le caractère se trouvant immédiatement après le marqueur de colonne `&`.
- Remarquer, dans le troisième exemple du tableau 4.9, comment 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}
```

$$\begin{aligned} a_1 &= b_1 + c_1 \\ a_2 &= b_2 + c_2 - d_2 + e_2 \end{aligned} \tag{4.13}$$

TAB. 4.9 – Comparaison des environnements pour les équations hors paragraphe de **amsmath** (les lignes verticales indiquent les marges logiques).

```
\begin{equation*}
  a = b
\end{equation*}
```

$$a = b$$

```
\begin{equation}
  a = b
\end{equation}
```

$$a = b \tag{4}$$

```
\begin{equation}
  \label{xx}
  \begin{split}
    a &= b + c - d \\
    &\phantom{=} + e - f \\
    &= g + h \\
    &= i
  \end{split}
\end{equation}
```

$$\begin{aligned} a &= b + c - d \\ &\quad + e - f \\ &= g + h \\ &= i \end{aligned} \tag{5}$$

```
\begin{multline}
  a + b + c + d + e + f \\
  + i + j + k + l + m + n
\end{multline}
```

$$\begin{aligned} &a + b + c + d + e + f \\ &\quad + i + j + k + l + m + n \end{aligned} \tag{6}$$

```
\begin{gather}
  a_1 = b_1 + c_1 \\
  a_2 = b_2 + c_2 - d_2 + e_2
\end{gather}
```

$$a_1 = b_1 + c_1 \tag{7}$$
$$a_2 = b_2 + c_2 - d_2 + e_2 \tag{8}$$

```
\begin{align}
  a_1 &= b_1 + c_1 \\
  a_2 &= b_2 + c_2 - d_2 + e_2
\end{align}
```

$$a_1 = b_1 + c_1 \tag{9}$$
$$a_2 = b_2 + c_2 - d_2 + e_2 \tag{10}$$

```
\begin{align}
  a_{11} &= b_{11} &
  a_{12} &= b_{12} \\
  a_{21} &= b_{21} &
  a_{22} &= b_{22} + c_{22}
\end{align}
```

$$a_{11} = b_{11} \quad a_{12} = b_{12} \tag{11}$$
$$a_{21} = b_{21} \quad a_{22} = b_{22} + c_{22} \tag{12}$$

Conseil du TEXpert

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-elles aux équations.
2. Lorsqu'une équation s'étend sur plus d'une ligne, couper chaque ligne *avant* un opérateur de sorte que chaque ligne constitue une expression mathématique complète (voir les troisième et quatrième exemples du tableau 4.9).
3. Ne numéroter que les équations d'un document auxquelles le texte fait référence.

- Les renvois vers des équations numérotées fonctionnent, comme partout ailleurs en LaTeX, avec le système d'étiquettes et de références (section 3.4). Le paquetage **amsmath** fournit également la pratique commande `\eqref` qui place automatiquement le numéro d'équation entre parenthèses.

```
On voit en \eqref{xx} du
tableau 4.9 que...
```

On voit en (5) du tableau 4.9 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.

`amsldoc` Consulter le chapitre 3 de la documentation du paquetage **amsmath** pour les détails sur l'utilisation des environnements du tableau 4.9.

Exemple 4.3. Les deux suites d'équations au début de la section ont été réalisées 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*}
```

□

4.6 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, non pas

$$(1 + \frac{1}{x}),$$

mais plutôt

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

La paire de commandes

```
\left⟨delim_g⟩  ...  \right⟨delim_d⟩
```

définit un délimiteur gauche ⟨*delim_g*⟩ et un délimiteur droit ⟨*delim_d*⟩ dont la taille s'ajustera automatiquement au contenu entre les deux commandes.

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

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

```
\left(
  \sum_{i = 1}^n x_i^2
\right)^{1/2}
```

$$\left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$$

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]
```

$$\left[\int \left(1 + \frac{x}{k}\right) dx\right]$$

- Les symboles ⟨*delim_g*⟩ et ⟨*delim_d*⟩ 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
```

$$\int_0^1 x\,dx = \left[\frac{x^2}{2}\right|_0^1$$

- 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{aligned}
    1 - x, &\quad x < 1 \\
    x - 1, &\quad x \geq 1
  \end{aligned}
\right.
```

$$f(x) = \left\{\begin{aligned} 1 - x, &\quad x < 1 \\ x - 1, &\quad x \geq 1 \end{aligned}\right.$$

(L'environnement `aligned` utilisé ci-dessus provient de **amsmath**.) On notera au passage que 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

TAB. 4.10 – Tailles des délimiteurs mathématiques

taille standard	( )	[ ]	{ }
\big	( )	[ ]	{ }
\Big	( )	[ ]	{ }
\bigg	( )	[ ]	{ }
\Bigg	( )	[ ]	{ }

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}{f} - g
    \right)
\end{align*}
```

$$a = \left(b + \frac{c}{d} \right. \\ \left. + \frac{e}{f} - g\right)$$

- Quand les choix de taille de délimiteurs de LaTeX ne conviennent pas pour une raison ou pour une autre, on peut sélectionner soi-même leur taille 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 4.10 contient des exemples de délimiteurs pour chaque taille.

La section 14 de la documentation de **amsmath** traite de divers enjeux typographiques en lien avec les délimiteurs et on y introduit des nouvelles commandes pour contrôler leur taille. C'est une lecture suggérée. amsldoc

Exemple 4.4. Le développement de la formule d'approximation de Simpson

comporte plusieurs des éléments discutés jusqu'à maintenant :

$$\begin{aligned}
\int_a^b f(x)\,dx &\approx \sum_{j=0}^{n-1} \int_{x_{2j}}^{x_{2(j+1)}} f(x)\,dx \\
&= \frac{h}{3} \sum_{j=0}^{n-1} \left[f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2(j+1)}) \right] \\
&= \frac{h}{3} \left[f(x_0) + \sum_{j=1}^{n-1} f(x_{2j}) + 4 \sum_{j=0}^{n-1} f(x_{2j+1}) \right. \\
&\quad \left. + \sum_{j=0}^{n-2} f(x_{2(j+1)}) + f(x_{2n}) \right] \\
&= \frac{h}{3} \left[f(a) + 2 \sum_{j=1}^{n-1} f(x_{2j}) + 4 \sum_{j=0}^{n-1} f(x_{2j+1}) + f(b) \right].
\end{aligned}$$

On compose 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. \\
  &\phantom{=} + \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{align*}
```

□

4.7 Caractères gras en mathématiques

Les caractères gras sont parfois utilisés en mathématiques, particulièrement pour représenter les vecteurs et les matrices :

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow \boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

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{⟨symbole⟩}
```

place ⟨*symbole*⟩ 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{+}
A \mathbf{A}
```

$\theta\boldsymbol{\theta} + \Gamma\boldsymbol{\Gamma} + A\mathbf{A}$

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{⟨symbole⟩}
```

fournie par le paquetage **bm** (Carlisle, 2014).

```
\theta \bm{\theta} +
\Gamma \bm{\Gamma} \bm{+}
A \bm{A}
```

$\theta\boldsymbol{\theta} + \Gamma\boldsymbol{\Gamma} + A\boldsymbol{A}$

Conseil du TEXpert

Si le gras est fréquemment utilisé dans un document pour une notation particulière, il est fortement recommandé de définir une nouvelle commande[3] 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, on pourrait définir une nouvelle commande `\mat` en insérant dans le préambule du document

```
\newcommand[1]{\mat}{\bm{#1}}
```

Les utilisateurs de XeLaTeX devraient charger le paquetage **unicode-math** (Robertson et collab., 2014) pour sélectionner leur police de caractère pour les mathématiques. Ce paquetage fournit la commande

```
\symbf{⟨symbole⟩}
```

pour placer un ⟨*symbole*⟩ 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.

unicode-math

```
% XeLaTeX + paquetage unicode-math
% avec l'option bold-style=ISO
\theta \symbf{\theta} +
\Gamma \symbf{\Gamma} \symbf{+}
A \symbf{A}
```

$$\theta\boldsymbol{\theta} + \Gamma\boldsymbol{\Gamma} + A\boldsymbol{A}$$

Exemple 4.5. 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.

Supposons que la commande `\mat` mentionnée dans la rubrique Conseil du TEXpert est définie dans le préambule du document. Alors le code source

```
\begin{align*}
  \mat{J}(\mat{\theta})
```

3. La définition de nouvelles commandes est couvert plus en détail au chapitre 6.

On l'a vu ci-dessus : 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 de 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 $\mathrm{i} = \sqrt{-1}$. Même chose pour les opérateurs mathématiques, dont le d dans les dérivées et les intégrales.
Voici un exemple d'intégrale composée avec ces régles :

$$\int_C \frac{\mathrm{e}^z}{z}\,\mathrm{d}z = 2\pi\mathrm{i}.$$

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.

```
  &=
  \begin{bmatrix}
    \dfrac{\partial f_1(\mat{\theta})}{\partial \theta_1} &
    \dfrac{\partial f_1(\mat{\theta})}{\partial \theta_2}
    \\[12pt] % augmenter l'espace entre les lignes
    \dfrac{\partial f_2(\mat{\theta})}{\partial \theta_1} &
    \dfrac{\partial f_2(\mat{\theta})}{\partial \theta_2}
  \end{bmatrix} \\
  &=
  \left[
  \frac{\partial f_i(\mat{\theta})}{\partial \theta_j}
  \right]_{2 \times 2}, \quad i, j = 1, 2.
\end{align*}
```

produit l'équation

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{\theta}) &= \begin{bmatrix} \dfrac{\partial f_1(\boldsymbol{\theta})}{\partial \theta_1} & \dfrac{\partial f_1(\boldsymbol{\theta})}{\partial \theta_2} \\ \dfrac{\partial f_2(\boldsymbol{\theta})}{\partial \theta_1} & \dfrac{\partial f_2(\boldsymbol{\theta})}{\partial \theta_2} \end{bmatrix} \\ &= \left[\frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_j} \right]_{2\times 2}, \quad i,j = 1,2.\end{aligned}$$

On remarquera l'utilisation de la commande `\dfrac` (section 4.3.2) pour composer des grandes fractions à l'intérieur des matrices. □

4.8 Exercices

4.1 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
>
> $$[f(x)g(x)]'' = f''(x)g(x) + 2f'(x)g'(x) + f(x)g''(x).$$

4.2 Composer l'équation suivante avec l'environnement `align*` :

$$\begin{aligned} f(x+h, y+k) = f(x,y) &+ \left\{ \frac{\partial f(x,y)}{\partial x} h + \frac{\partial f(x,y)}{\partial y} k \right\} \\ &+ \frac{1}{2} \left\{ \frac{\partial^2 f(x,y)}{\partial x^2} h^2 + \frac{\partial^2 f(x,y)}{\partial x \partial y} kh + \frac{\partial^2 f(x,y)}{\partial y^2} k^2 \right\} \\ &+ \frac{1}{6}\{\cdots\} + \cdots + \frac{1}{n!}\{\cdots\} + R_n. \end{aligned}$$

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.

4.3 Composer à l'aide de l'environnement `cases` (section 4.6) la définition de la fonction $\tilde{f}(x)$:

$$\tilde{f}(x) = \begin{cases} 0, & x \le c_0 \\ \dfrac{F_n(c_j) - F_n(c_{j-1})}{c_j - c_{j-1}} = \dfrac{n_j}{n(c_j - c_{j-1})}, & c_{j-1} < x \le c_j \\ 0, & x > c_r. \end{cases}$$

Il est nécessaire d'imposer la taille des fractions dans la seconde branche de la définition à l'aide des fonctions de la section 4.3.2.

4.4 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 XeLaTeX. Effectuer ensuite les modifications suivantes.

a) Charger le paquetage **amsfonts** dans le préambule, puis remplacer `$R^+$` par `$\mathbb{R}^+$` à la ligne débutant par « Le domaine ».

b) 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.

c) Toujours dans la même matrice, composer successivement les deux fractions avec les commandes `\frac`, `\tfrac` et `\dfrac`. Observer le résultat.

d) Réduire l'espacement de part et d'autre du symbole $\Leftrightarrow$ dans la seconde équation hors paragraphe.

e) À l'aide de la fonction Rechercher et remplacer de l'éditeur de texte, remplacer toutes les occurrences du symbole θ par λ.

5 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 mémoire, 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. Un auteur ne devrait se 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.

5.1 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.

5.1.1 BibTeX et natbib

Avec plus de 25 années d'utilisation, BibTeX (Patasnik, 1988) est le système standard de traitement des bibliographies dans LaTeX. Il est stable et

prévisible — ce que d'aucuns 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. Non qu'il s'agisse d'un argument massue, mais même Wikipedia, dans les rubriques « Citer cette page », offre les citations en format BibTeX.

BibTeX est principalement un système de tri d'entrées bibliographiques et d'interface avec la base de données. Contrôler l'apparence des citations et de la bibliographie requiert un *style*. Les styles standards sont `plain`, `unsrt`, `alpha` et `abbrv`; nous y reviendrons à la section 5.4.

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-année [1] fréquemment utilisé dans les sciences naturelles et sociales. Il 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. D'ailleurs, la classe `ulthese` pour les thèses et mémoires de l'Université Laval charge par défaut le paquetage.

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 and citations* de la formidable *UK List of TeX Frequently Asked Questions*.

`letterfaq`

5.1.2 Biber et biblatex

Au moment d'écrire ces lignes, un nouveau système de traitement des bibliographies dans LaTeX est en émergence. Il est formé du moteur de traitement Biber (Kline et Charette, 2015) et du paquetage **biblatex** (Lehman, 2015). 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

1. C'est le style utilisé dans le présent document.

saura s'établir comme nouveau standard, surtout compte tenu de la masse de matériel disponible pour BibTeX.

Pour de l'information additionnelle, consulter cette entrée du site `tex.stackexchange.com` qui fournit un excellent sommaire des mérites et des inconvénients respectifs des deux systèmes de traitement de bibliographie.

En l'absence d'un consensus clair, nous avons choisi de traiter dans ce chapitre à la fois du système le plus répandu et de celui avec lequel nous sommes le plus familier, soit la combinaison BibTeX et **natbib**.

5.1.3 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 des données bibliographiques de EndNote. Une simple recherche dans Internet sur « EndNote BibTeX » devrait fournir toute l'information nécessaire pour réaliser la conversion.

5.2 Processus de création d'une bibliographie

La création d'une bibliographie compte plusieurs étapes. Nous les présentons 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. On utilise les mêmes bases de données pour tous nos documents. Par conséquent, le temps consacré à cette étape s'amenuise au fur et à mesure que l'on complète nos bases de données.
2. Choisir un style de citation et de présentation de la bibliographie, généralement en se fiant aux us et coutumes dans sa discipline scientifique.
3. Insérer dans le texte de son document des références à des ouvrages se trouvant dans ses bases de données bibliographiques.
4. Insérer dans le code source du document les informations relatives au style de bibliographie et aux bases de données à utiliser, puis composer les références et la bibliographie avec BibTeX.

5.3 Création d'une base de données

Il est tout à fait possible de citer des références et de construire une bibliographie avec LaTeX sans avoir recours à une base de données bibliographiques et à BibTeX pour traiter celles-ci. L'investissement requis en temps et en efforts pour adopter l'approche BibTeX demeure toutefois faible, surtout au regard des avantages :

- on entre les informations dans une base de données une seule fois pour ensuite les utiliser à répétition ;
- le traitement automatisé des informations assure une présentation uniforme de celles-ci ;
- on peut changer le style de présentation de la bibliographie 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 5.1.

Exemple 5.1. 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 (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 =      {978-0321173850},
  language =  {english}
}
```

Kopka, H. et P. W. Daly. 2003, *Guide to LaTeX*, 4e éd., Addison-Wesley, ISBN 978-0321173850.

```
@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 LaTeX», *TUGboat*, vol. 30, no 1, p. 36-48. URL https://www.tug.org/TUGboat/tb30-1/tb94mori.pdf.

```
@Manual{natbib,
  author =   {Patrick W. Daly},
  title =    {Natural Sciences Citations and References},
  year =     2010,
  url =      {http://www.ctan.org/pkg/natbib/},
  language = {english}
}
```

Daly, P. W. 2010, *Natural Sciences Citations and References*. URL `http://www.ctan.org/pkg/natbib/`.

□

btxdoc

- Les types d'entrée bibliographique dans l'exemple ci-dessus sont `Book`, `Article` et `Manual`[2]. On remarquera que les champs utilisés sont différents d'un type à un autre.

  On trouvera la liste de tous les types d'entrée et des champs obligatoires et optionnels pour chacun dans, entre autres, Wikipedia, la documentation de BibTeX ou la plupart des bons ouvrages de référence (comme Kopka et Daly, 2003).
- On entre le nom d'un auteur 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 le mot-clé `and`.
- BibTeX gère normalement 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 5.1, 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` et quelques autres (voir 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 césure 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 « 2e édition » ou « 2nd edition » selon que l'on a précisé que l'ouvrage est en français ou en anglais.

2. Les identifiants des types d'entrée et des champs sont insensibles à la casse. Par exemple, on pourrait tout aussi bien débuter une entrée par `@Manual`, `@manual` ou `@MANUAL`.

Conseil du TEXpert

Tenter d'entretenir une grande base de données bibliographiques unique peut rapidement devenir pénible.
Mieux vaut scinder ses références dans plusieurs fichiers par thématique, un peu comme dans une bibliothèque : informatique, mathématiques, statistique, droit, etc.
Nommer ensuite les fichiers du nom de la thématique : `informatique.bib`, `droit.bib`, etc.

- BibTeX supportera les lettres accentuées ou autres caractères spéciaux dans le texte des champs seulement si les paquetages requis pour entrer ces caractères dans le texte sont chargés (**inputenc** avec pdfLaTeX ; **fontspec** avec XeLaTeX). Autrement, il faut entrer ceux-ci avec les commandes LaTeX de base ; voir le tableau 17 de la Comprehensive LaTeX Symbol List pour les accents. `comprehensive`

  Notre recommandation : éviter les lettres accentuées dans les entrées susceptibles d'être utilisées dans un document entièrement en anglais.

5.4 Style des citations et de la bibliographie

Il existe plusieurs manières différentes de présenter une bibliographie et LaTeX s'adapte tout naturellement aux divers besoins des utilisateurs. 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-année, etc.) ;
2. la présentation des notices bibliographiques (ordre des éléments, ponctuation, mise en forme des caractères, etc.).

On trouvera des exemples de quelques styles de bibliographie dans le tableau 5.1.

Tel que mentionné précédemment, 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, nous recommandons d'utiliser le paquetage **natbib** pour la gestion des références et du style de la bibliographie. Entre

TAB. 5.1 – Quelques styles de bibliographie 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. *Guide to LaTeX*. Addison-Wesley, 4 edition, 2003.
`plain-fr`	Un bon ouvrage de référence est [1]	[1] Helmut KOPKA et Patrick W. DALY : *Guide to LaTeX*. Addison-Wesley, 4 édition, 2003.
`alpha-fr`	Un bon ouvrage de référence est [KD03]	[KD03] Helmut KOPKA et Patrick W. DALY : *Guide to LaTeX*. Addison-Wesley, 4 édition, 2003.

Styles auteur-année 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 : *Guide to LaTeX*. Addison-Wesley, 4 édition, 2003.
`francais`	Un bon ouvrage de référence est Kopka et Daly (2003)	Kopka, H. et P. W. Daly. 2003. *Guide to LaTeX*, 4e éd., Addison-Wesley.

autres choses, ce paquetage supporte le style de citation auteur-année 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 et les URL. Le paquetage fournit des styles `plainnat`, `unsrtnat` et `abbrvnat` similaires aux styles standards, mais plus complets. Il existe des versions francisées de ces styles (et de quelques autres) dans CTAN et dans TeX Live. Il est fortement recommandé de consulter la documentation de **natbib** pour les détails. On y trouvera également des informations sur l'utilisation de styles de citation autres que auteur-année.

`natbib`

On trouve également dans CTAN et dans TeX Live le paquetage **francais-bst** (Goulet, 2013) qui fournit deux feuilles de style compatibles avec **natbib**. Celles-ci permettent de composer des bibliographies auteur-année respectant les normes de typographie française proposées dans Malo (1996) ; voir le dernier exemple du tableau 5.1.

Le paquetage **natbib** est chargé par défaut par la classe ulthese ; consulter la section 6.6 de la documentation de la classe pour plus d'information sur l'interaction entre le paquetage et celle-ci.

ulthese

En terminant, notons que la plupart des journaux scientifiques et des maisons d'édition ont leur propre style que les auteurs sont tenus d'utiliser. En ce qui a trait aux thèses et mémoires de l'Université Laval, la Faculté des études supérieures et postdoctorales n'impose aucun style de bibliographie particulier. Nous recommandons d'utiliser le style usuel dans sa discipline scientifique.

5.5 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 LaTeX 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 sa bibliographie.

Avec **natbib**, on utilisera plutôt les commandes

```
\citet{⟨clé⟩}
\citep{⟨clé⟩}
```

Dans le style de citation auteur-année, ces commandes permettent respectivement d'insérer une référence au fil de la phrase ou en aparté :

```
\citet{Mori:bibliographies:2009}
en offre un bon survol.
```

Mori (2009), en offre un bon survol.

```
TUGboat a publié un bon survol
\citep{Mori:bibliographies:2009}.
```

TUGboat a publié un bon survol (Mori, 2009).

On ne devrait *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, on devrait utiliser les commandes de **natbib**

```
\citeauthor{⟨clé⟩}
\citeyear{⟨clé⟩}
```

Ainsi, 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.

Le paquetage fournit plusieurs autres commandes pour manipuler les informations bibliographiques et contrôler leur présentation ; consulter la documentation de **natbib**.

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, insérer dans le corps du document la commande

```
\nocite{⟨clé1,clé2,...⟩}
```

où ⟨*clé1*⟩, ⟨*clé2*⟩, …, sont les clés des documents à inclure dans la bibliographie.

5.6 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.

Le processus de création d'un document avec pdfLaTeX ou XeLaTeX se représente schématiquement ainsi (Goulet, 2015) :

Pour créer ou mettre à jour la bibliographie, il s'ajoute au processus une étape de compilation du document avec BibTeX :

Conseil du TEXpert

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.
Il peut arriver que l'hyperlien soit superflu ou indésirable. Pour le supprimer pour une référence particulière, on utilise l'environnement `NoHyper` :

```
\begin{NoHyper} \citet{⟨clé⟩} \end{NoHyper}
```

Pour usage fréquent, définir une nouvelle commande (chapitre 6). Par exemple, avec dans le préambule

```
\newcommand{\nolink}[1]{%
  \begin{NoHyper}#1\end{NoHyper}}
```

on pourra utiliser dans le texte

```
\nolink{\citet{⟨clé⟩}}
```

pour supprimer l'hyperlien d'une référence bibliographique.

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. Ajouter dans le préambule ou près de la commande `\bibliography` ci-dessous la commande

   ```
   \bibliographystyle{⟨style⟩}
   ```

   où ⟨*style*⟩ est un nom de style bibliographique.
3. Ajouter dans le texte la commande

   ```
   \bibliography{⟨base_donnees1⟩, ⟨base_donnees2⟩, ...}
   ```

   à l'endroit où l'on veut qu'apparaisse la bibliographie (généralement à la fin du document). Les arguments ⟨*base_donnees1*⟩, ⟨*base_donnees2*⟩, séparés par des virgules, sont les noms (sans l'extension `.bib`) des fichiers de données bibliographiques.

Conseil du TEXpert

Aux toutes dernières étapes avant de rendre un document, s'assurer d'exécuter BIBTEX une dernière fois et de compiler avec pdfLATEX ou XELATEX au moins deux fois. Le journal de la compilation (*log file*) ne devrait pas rapporter de références manquantes (*undefined references*).

4. Compiler le document une première fois avec un moteur TEX 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 BIBTEX afin de préparer le texte des références et composer la bibliographie.
6. Compiler à nouveau le document au moins deux fois avec un moteur TEX 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. Autrement, tant que la bibliographie demeure inchangée, une compilation standard avec seulement le moteur TEX suffit.

Exemple 5.2. La figure 5.1 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-année de **natbib**.
- La présence du paquetage **fontspec** dans le préambule indique que le document doit être compilé avec XELATEX. 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 du paquetage, le français est la langue par défaut du document.

Ces fichiers en main, les étapes de composition 1–3 sont complétées. La figure 5.2 présente la zone de texte principale du document après l'étape 4, puis après chacune des deux compilations de l'étape 6. □

Les éditeurs de texte spécialisés pour la préparation de documents LATEX offrent généralement des raccourcis pour exécuter la compilation avec BIBTEX.

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]{article}
  \usepackage{fontspec}
  \usepackage{natbib}
  \usepackage[english,french]{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. 5.1 – 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*, 4e 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*, 4e 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. 5.2 – Zone de texte du document aux diverses étapes de la compilation des fichiers de la figure 5.1 avec XeLaTeX et BibTeX

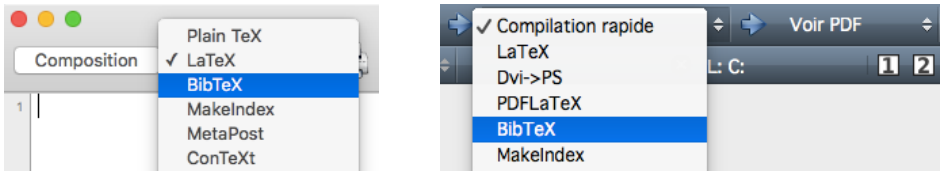


FIG. 5.3 – Interfaces de sélection du programme BIBTEX dans TeXShop (à gauche) et Texmaker (à droite)

- Dans TeXShop, on sélectionne un autre programme dans le menu à côté du bouton « Composition ».
- Dans Texmaker, on choisit le programme approprié dans le menu de composition rapide.

Ces deux interfaces sont représentées à la figure 5.3.

On trouvera des informations additionnelles, notamment sur des sources de données bibliographiques et des outils de gestion des bases de données, dans la section Gestion de la bibliographie de Wikilivres.

5.7 Exercices

5.1 Composer des entrées de base de données pour les références bibliographiques suivantes.

a) Mittelbach, F. 2014, «How to influence the position of float environments like figure and table in LATEX?», *TUGboat*, vol. 35, nº 3, p. 258–254. URL `https://www.tug.org/TUGboat/tb35-3/tb111mitt-float.pdf`

Astuce : cette entrée est un article tiré d'une revue scientifique.

b) Wilson, P. 2013, *The Memoir Class for Configurable Typesetting*, 8e éd., The Herries Press. URL `http://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 TEX*. URL `http://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`.

5.2 Utiliser pour cet exercice le fichier `exercice_gabarit.tex` ainsi que la base de données bibliographique crée 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 5.6 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 `francais` fourni par **francais-bst** (installé par défaut dans TeX Live). Recompiler le document et observer les différences par rapport aux documents produit en a).

5.3 À partir d'un gabarit fourni avec la classe ulthese, produire un document simple contenant une bibliographie.

6 Commandes et environnements définis par l'usager

LaTeX est un ensemble de macro commandes conçu pour faciliter l'utilisation du système TeX. Dès lors, il est assez naturel de permettre à l'usager de définir à son tour ses propres 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 6.1. Nous avons créé ou modifié des commandes pour chacune des raisons ci-dessus dans la préparation du présent document.

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 nous souhaitons en changer, il suffit de modifier la définition de la commande.

3. La redéfinition de la commande `\chaptitlefont` de la classe memoir permet de modifier la police de caractère et la mise en forme des titres de chapitres.

Nous reviendrons sur les détails de ces exemples dans la suite du chapitre. □

6.1 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 les appels à ces commandes dans le préambule du document.

6.1.1 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{\⟨nom_commande⟩}{⟨définition⟩}
\renewcommand{\⟨nom_commande⟩}{⟨définition⟩}
```

Le premier argument, \⟨*nom_commande*⟩, est le nom de la commande, avec le caractère \. Ce nom doit être différent de celui de toute commande active [1] dans le document dans le cas de `\newcommand`. À l'inverse, une commande active doit nécessairement porter le même nom lorsque l'on fait appel à `\renewcommand`.

Le second argument, ⟨*définition*⟩, 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 6.2. La commande `\mathbb`, présentée à la section 4.4.2, 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}}
```

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.

Ainsi, si l'on souhaite pour une raison quelconque modifier la représentation de l'ensemble des nombres réels, il suffit de modifier la définition de la commande \R pour que le changement prenne effet dans tout le document. □

Exemple 6.3. Tel que mentionné à l'exemple 6.1, nous avons modifié la police de caractère des titres de chapitres dans le présent document en redéfinissant la commande \chaptitlefont de la classe memoir. Pour obtenir des titres de chapitres sans empattements, en caractères gras, de dimension \Huge et alignés à gauche, on trouve dans le préambule du document la déclaration

```
\renewcommand{\chaptitlefont}{\normalfont%
  \sffamily\bfseries\Huge\raggedright}
```

On trouve la commande \normalfont au début de la définition à titre préventif afin de « remettre à zéro » toutes les caractéristiques de police. □

6.1.2 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 6.4. La nouvelle commande \pkg mentionnée à l'exemple 6.1 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 6.5. La commande \doc mentionnée à l'exemple 6.1 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 de caractère non proportionnelle ;
3. l'URL vers le fichier de documentation en ligne.

Une version simplifiée de la définition de la commande est la suivante :

```
\newcommand{\doc}[3]{%
  \href{#3}{#1~\raisebox{-0.2ex}{\faExternalLink}}%
  \marginpar{\faBook~\texttt{#2}}}
```

La commande `\href` qui permet d'insérer un hyperlien dans le texte provient du paquetage **hyperref** (Rahtz et Oberdiek, 2012). Les commandes `\faBook` et `\faExternalLink` proviennent du paquetage **fontawesome** (Danaux et Gandy, 2015) ; elles insèrent dans le texte des icônes de la police libre Font Awesome.

Avec la définition ci-dessus, la déclaration

```
\doc{documentation}{hyperref}{%
  http://texdoc.net/pkg/hyperref}
```

hyperref produit : documentation. □

6.2 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{⟨...⟩}` ; le contenu en tant que tel ; la fin, marquée par la déclaration `\end{⟨...⟩}`.

On crée ou modifie des environnements avec les commandes

```
\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.

⟨*fin_déf*⟩ commandes et texte à insérer à la fin de l'environnement, lors de l'appel `\end{`⟨*nom_env*⟩`}`.

Exemple 6.6. On souhaite 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 `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}
```

□

Exemple 6.7. Nous avons créé pour les fins du présent document un environnement `conseil` servant à mettre en forme les rubriques « Conseil du TEXpert ». La définition — relativement élaborée — de l'environnement est la suivante :

```
\newenvironment{conseil}{%
  \colorlet{TFFrameColor}{black}
  \colorlet{TFTitleColor}{white}
  \begin{table}
    \begin{titled-frame}{\sffamily Conseil du {\TeX}pert}
```

Conseil du TEXpert

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.

```
      \noindent
      \begin{minipage}{0.1\linewidth}
        \raisebox{-1.5em}[0em][0em]{\HUGE\faThumbsOUp}
      \end{minipage}
      \begin{minipage}[t]{0.88\linewidth}}%
  {\end{minipage}\end{titled-frame}\end{table}}
```

Dans le second argument, on :

1. définit des couleurs requises par l'environnement `titled-frame` ;
2. ouvre un élément flottant pour disposer la rubrique sur la page ;
3. ouvre un environnement `titled-frame` (fourni par le paquetage **framed**) pour encadrer la rubrique et afficher son titre ;
4. crée une première `minipage` pour disposer le symbole 👍 tiré de la police Font Awesome ;
5. ouvre une seconde `minipage` à côté de la première pour accueillir le texte de la rubrique.

Le troisième argument sert à refermer tous les environnements ouverts dans le second argument et qui n'ont pas déjà été fermés.

Une fois ce travail accompli, créer une nouvelle rubrique est très simple :

```
\begin{conseil}
  N'hésitez pas à créer des nouvelles commandes...
\end{conseil}
```

On trouvera le résultat ci-dessus. □

6.3 Exercices

6.1 Certains auteurs composent les sigles et les acronymes [2] en petites capitales, avec ou sans les points : C.Q.F.D., NASA.

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.

a) Créer une commande `\NASA` qui insère l'acronyme NASA dans le texte. Rappelons que l'on compose du texte en petites capitales avec la commande `\textsc`.

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.

6.2 Nous n'avons 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{\⟨nom_commande⟩}[⟨narg⟩][⟨option⟩]{⟨définition⟩}
\renewcommand{\⟨nom_commande⟩}[⟨narg⟩][⟨option⟩]{⟨définition⟩}
```

L'argument additionnel ⟨*option*⟩ contient la valeur par défaut du *premier* argument de \⟨*nom_commande*⟩. On remarquera que, dès lors, la commande ne compte plus ⟨*narg*⟩ arguments obligatoires, mais bien ⟨*narg*⟩ − 1 arguments obligatoires et un optionnel.

Modifier la définition de la commande `\doc` de l'exemple 6.5 pour que « documentation » soit le texte par défaut de l'hyperlien qui est placé au fil du texte.

6.3 Modifier l'environnement `citation` de l'exemple 6.6 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.

7 Trucs et astuces divers

7.1 Polices de caractères : au-delà de Computer Modern

Les documents LaTeX standards sont facilement reconnaissables par leur police de caractères 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 de caractères. Fort heureusement, la situation a beaucoup évolué et il est aujourd'hui assez simple de produire des documents LaTeX utilisant des polices de caractères variées.

Avant d'aller plus loin, une mise en garde : si votre document contient plus que quelques équations mathématiques très simples, le choix de police devient très restreint. La raison : peu de polices de caractères comprennent des symboles mathématiques et les informations nécessaires pour les assembler dans une mise en page satisfaisant les standards de haute qualité usuels de LaTeX.

Cela dit, pour qui souhaite aller au-delà de la police Computer Modern sans trop se compliquer la vie, il existe deux solutions principales.

1. Utiliser l'une ou l'autre des polices PostScript standards. C'est très simple avec toute distribution LaTeX moderne : il suffit de charger le paquetage approprié. Consulter la documentation de l'ensemble de paquetages PSNFSS pour connaître les choix disponibles. `psnfss2e`
2. Utiliser une police OpenType présente sur son système avec le moteur XeLaTeX. Seule une poignée de ces polices offrent toutefois un support approprié pour les mathématiques. La gestion des polices de caractères avec XeLaTeX se fait avec le paquetage standard **fontspec** ; consulter sa documentation. `fontspec`

Pour les thèses et mémoires de l'Université Laval, la Faculté des études supérieures et postdoctorales accepte les polices

Computer Modern	ABCDEF abcdef 1234567890
Times	ABCDEF abcdef 1234567890
Palatino	ABCDEF abcdef 1234567890

Pour utiliser ces deux dernières avec LaTeX, on charge respectivement les paquetages **mathptmx** ou **mathpazo**. Avec XeLaTeX, on utilisera les polices Termes et Pagella du projet TeX Gyre. Ce sont des polices très similaires à Times et Palatino, disponibles en version OpenType et qui fournissent un bon support pour les mathématiques via le projet frère TeX Gyre Math.

Exemple 7.1. Pour utiliser la police PostScript classique Palatino avec LaTeX tant pour le texte que pour les mathématiques, il suffit d'insérer dans le préambule de son document la commande

```
\usepackage{mathpazo}
```

Avec le moteur XeLaTeX, il est possible d'utiliser n'importe quelle police de caractères OpenType et TrueType installée dans le système d'exploitation de l'ordinateur. Pour obtenir un résultat équivalent à celui de **mathpazo**, on installe les polices TeX Gyre dans le système, puis on insère dans le préambule les commandes

```
\usepackage{fontspec}
\setmainfont{TeX Gyre Pagella}
\setmathfont{TeX Gyre Pagella Math}
```

□

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 produrer une copie gratuitement en écrivant à lucida@bibl.ulaval.ca.

7.2 Couleurs

xcolor

L'utilisation de couleur dans un document LaTeX requiert de charger le paquetage **xcolor** (Kern, 2007). Celui-ci définit d'abord plusieurs couleurs que l'on peut utiliser directement ; consulter la documentation pour en connaître les différentes listes. Le tableau 7.1 fournit celle des couleurs toujours disponibles.

On modifie la couleur du texte avec la commande

TAB. 7.1 – Couleurs toujours disponibles quand le paquetage **xcolor** est chargé

black	blue	brown	cyan	darkgray
gray	green	lightgray	lime	magenta
olive	orange	pink	purple	red
teal	violet	white	yellow	

```
\color{⟨nom⟩}
```

où ⟨*nom*⟩ est le nom d'une couleur. À moins de vouloir modifier la couleur de tout ce qui suit, on limitera la portée de la commande par des accolades.

```
texte {\color{red} en rouge}
et {\color{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) où 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 7.2. La commande

```
\definecolor{acier}{rgb}{0.1,0.4,0.6}
```

définit une nouvelle couleur nommée `acier` composée de rouge 30 %, de vert 40 % et de bleu 60 % : + + = . On pourra utiliser la couleur `acier` directement dans la commande `\color`. □

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.

7.3 Hyperliens et métadonnées de documents PDF

Nous en avons déjà traité à quelques reprises, notamment à la section 4 de Goulet (2015) : le paquetage **hyperref** (Rahtz et Oberdiek, 2012) permet de transformer toutes les références dans le texte en hyperliens cliquables lorsque le document est produit avec pdfLaTeX ou XeLaTeX. C'est très pratique lors de la consultation électronique d'un document.

Le paquetage offre une multitudes d'options de configuration ; nous n'en présenterons 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 l'on pourra souhaiter configurer, c'est le comportement et la couleur des divers types d'hyperliens. On trouvera ci-dessous les options de configuration pertinentes, leur valeur (avec en gras la valeur par défaut) ainsi qu'une brève explication de chacune.

`colorlinks`	`true`|**`false`**	colorer les liens selon leur type
`linktocpage`	`true`|**`false`**	faire du numéro de page plutôt que du titre l'hyperlien dans la table des matières
`linkcolor`	⟨*couleur*⟩	couleur des liens internes
`urlcolor`	⟨*couleur*⟩	couleur des URL externes
`citecolor`	⟨*couleur*⟩	couleur des citations
`allcolor`	⟨*couleur*⟩	couleur pour tous les types d'hyperliens

Lorsque la valeur admissible d'une option est `true` ou `false`, sa seule mention équivaut à `true`. La valeur ⟨*couleur*⟩ est le nom d'une couleur telle que définie par **xcolor** (section 7.2).

Les fichiers PDF peuvent contenir diverses métadonnées sur leur contenu. Le paquetage **hyperref** permet de définir certaines catégories de celles-ci, notamment les informations de document comme le titre ou l'auteur. Nous ne mentionnons ici que deux options de configutation ; la section 3.7 de la documentation contient la liste complète.

`hyperref`

`pdftitle`	texte	titre du document PDF
`pdfauthor`	texte	auteur du document PDF

Exemple 7.3. Le présent document fait appel aux définitions de couleurs et aux options de configuration de **hyperref** suivantes :

```
\definecolor{link}{rgb}{0,0.4,0.6}   % ~RoyalBlue de dvips
```

Conseil du TEXpert

L'interaction du paquetage **hyperref** avec les autres est parfois (voire souvent) délicate. Pour cette raison, il est fortement recommandé de charger **hyperref** en tout dernier dans dans le préambule.

```
\definecolor{url}{rgb}{0.6,0,0}      % rouge-brun
\definecolor{citation}{rgb}{0,0.5,0} % vert foncé
\hypersetup{colorlinks, linktocpage,
  linkcolor=link, urlcolor=url, citecolor=citation,
  pdftitle={Rédaction de thèses et mémoires avec LaTeX -
            Partie II},
  pdfauthor={Université Laval}}
```

□

Exemple 7.4. Les gabarits de thèses et de mémoires livrés avec la classe ulthese contiennent dans le préambule la ligne suivante :

```
\hypersetup{colorlinks,allcolors=ULlinkcolor}
```

Tous les liens de la thèse ou du mémoire seront donc de la même couleur, `ULlinkcolor` , une couleur définie par la classe. □

7.4 Présentation de code informatique

L'environnement standard `verbatim` de LaTeX permet de présenter du texte exactement tel qu'il est entré dans le code source du document. C'est un environnement particulièrement utile pour afficher du code informatique puisque le texte est composé en police non proportionnelle et que 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;
}
```

Si un document doit contenir beaucoup de code informatique et que l'on souhaite exercer un fin contrôle sur sa disposition et sa mise en forme, il vaut mieux se tourner vers un paquetage spécialisé comme **listings** (Heinz et collab., 2015). La documentation du paquetage compare ses fonctionnalités à celles de plusieurs autres paquetages similaires.

listings

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, en utilisant l'environnement `lstlisting` de **listings** plutôt que `verbatim`, l'extrait de code C ci-dessus pourrait se présenter ainsi :

```
\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 document d'expliquer les nombreuses fonctionnalités de **listings**. Précisons simplement que le paquetage a servi pour en composer les extraits de code LaTeX et pour construire une grande partie de l'index.

Exemple 7.5. 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}{0.7,0,0}

%% Configuration de listings
\lstset{language=[LaTeX]TeX,
  basicstyle=\ttfamily\NoAutoSpacing,
  keywordstyle=\mdseries,
  commentstyle=\color{comments}\slshape,
  extendedchars=true,
  showstringspaces=false,
  backgroundcolor=\color{LightYellow1},
  frame=lr, rulecolor=\color{LightYellow1},
  xleftmargin=3.4pt, xrightmargin=3.4pt}
```

(La couleur `LightYellow1` est définie par **xcolor** lorsque le paquetage est chargé avec l'option `x11names`.) □

7.5 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 pour ce 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 ici 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 utilisant 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 placés entre les balises <<>>= et @ 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. On compile le fichier `.tex` 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 <<>>=. Consulter la documentation de Sweave pour les détails.

Pweave est un système similaire à Sweave pour le langage Python.

Inspiré de Sweave, knitr (Xie, 2016) 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, rappelons que Sweave n'est qu'un exemple de système de programmation lettrée. Il en existe plusieurs autres. Il appartient dès lors au lecteur de trouver le système qui correspond le mieux à ses 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.
Le lecteur intéressé pourra consulter, par exemple, le code source de la classe ulthese. La documentation de la classe, le code LaTeX, les gabarits, tout se trouve entrelacé dans le seul fichier `ulthese.dtx`.

7.6 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, qui conviendront à l'auteur qui ne recherche 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., 2015). Celle-ci compte un grand nombre de thèmes et de gabarits élaborés, rend très simple l'insertion d'animations d'une diapositive à l'autre, gère automatiquement la table des matières et… produit des diapositives en couleur !

Quelle que soit la classe utilisée, les diapositives produites avec LaTeX se présentent sous forme de fichier PDF. On les projète avec une liseuse PDF en mode plein écran.

beamer

Nous n'irons pas plus loin sur le sujet dans ce document. Consulter la documentation de beamer pour apprendre à utiliser la classe. Produire des diapositives de grande qualité avec les gabarits fournis avec la classe est simple et rapide.

Les diapositives de la première partie de la formation (Goulet, 2015) ont été produites avec beamer.

7.7 Gestion des versions et travail collaboratif

Vous travaillez à plusieurs sur un même fichier ou seul, mais de plusieurs postes de travail différents. Quelle est la plus récente version du fichier ? Mon ajout fait hier dans le fichier a-t-il été pris en compte par ma 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. Les systèmes les plus populaires en ce moment sont Subversion (Collins-Sussman et collab., 2008) et Git (Chacon et Straub, 2014).

Bien que développés à l'origine pour la gestion du code source de logiciels, les systèmes de contrôle 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 serveur central (obligatoire avec Subversion ; optionnel avec Git [1].)

Un système de gestion de versions est un outil qui permet d'augmenter considérablement sa productivité ou celle de son équipe de travail au moment de rédiger un ouvrage scientifique.

La page Wikipedia sur le sujet offre une bonne introduction aux systèmes de gestion de versions. Les membres de la communauté de l'Université Laval qui souhaitent mettre sur pied un dépôt pour un projet pourront consulter leur équipe de soutien informatique facultaire.

La présente documentation est conservée dans un dépôt Subversion public d'où l'on peut toujours obtenir la plus récente version. Consulter la page des notices de copyright au début du document pour accéder au dépôt.

7.8 Exercices

Pour les exercices 7.1–7.5, utiliser le fichier `exercices_trucs.tex`. Celui-ci reprend une partie de la documentation de la classe ulthese. De plus, on remarquera que le document :

- doit être compilé avec pdfLaTeX puisqu'il charge les paquetages **inputenc** et **fontenc** ;

1. Si ce n'est que pour le volet de sauvegarde, nous recommandons d'utiliser avec Git un dépôt central tel que GitHub, un serveur public qui héberge déjà des millions de projets.

- définit des nouvelles commandes `\class` et `\fichier` pour composer, respectivement, les noms de classe et les noms de fichier ;
- utilise la commande `\doc` de l'exemple 6.5 ;
- 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.

7.1 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 dans la police Palatino, tel qu'expliqué à l'exemple 7.1. Charger également le paquetage **helvet** afin d'utiliser Helvetica pour le texte en police sans empattements (`\textsf`).

7.2 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.

7.3 Charger le paquetage **xcolor** et ajouter l'option `colorlinks` à **hyperref**, puis recompiler le document. Observer les changements.

7.4 En s'inspirant de l'exemple 7.3, modifier la couleur des liens internes et externes.

7.5 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 7.5, configurer la présentation des extraits de code pour utiliser une police non proportionnelle (`\ttfamily`) et un arrière-plan de la couleur standard « lightgray ».

7.6 Accéder au dépôt qui héberge le code source de cette formation LaTeX en suivant le lien à la page des notices de copyright. Télécharger le fichier maître de la première partie de la formation et ses fichiers auxiliaires :

```
formation_latex-partie_1.tex
├── licence-partie_1.tex
├── colophon-partie_1.tex
├── exercice_commandes-output.pdf
├── renvoi.pdf
├── renvoi_avec_ref.pdf
├── renvoi_avec_autoref.pdf
└── ponctuation.pdf
```

(Les lecteurs curieux pourront aussi installer sur leur poste de travail un client Subversion et l'utiliser pour extraire l'ensemble des fichiers du dépôt.)

Une fois les fichiers en main, examiner le code source du fichier maître. Il s'agit d'un document utilisant la classe beamer pour créer des diapositives.

Désactiver (effacer ou mettre en commentaire) les lignes suivantes du préambule :

```
\defaultfontfeatures{Ligatures=TeX,Scale=0.92}
\setmainfont{Lucida Bright OT}
\setsansfont{Lucida Sans OT}
\setmathfont{Lucida Bright Math OT}
\setmonofont{Lucida Grande Mono DK}
\newfontfamily\titles{Myriad Pro}
```

et, plus loin dans le fichier,

```
\setbeamerfont{frametitle}{family=\titles}
```

Compiler ensuite le document avec XeLaTeX. En comparant le code source et les diapositives, observer comment beamer peut afficher des listes une puce à la fois à partir d'un environnement `itemize` usuel une fois qu'on lui a ajouté une option `[<+->]`.

A Solutions des exercices

Chapitre 2

2.1 Une première boîte verticale de 10 cm de large contient le texte :

```
\parbox{10cm}{Ce bloc [...] la ligne.}
```

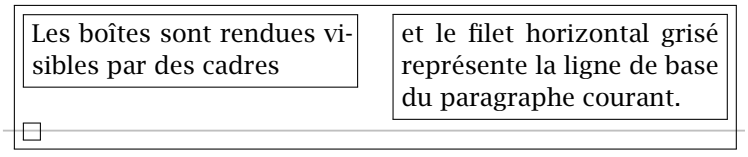
Cette boîte peut être placée dans une boîte horizontale encadrée avec `\fbox`. Celle-ci peut à son tour être placée dans une autre boîte horizontale encadrée, de manière à obtenir un cadrage double. Pour centrer le tout sur la ligne, on a recours à l'environnement `center` :

```
\begin{center}
  \fbox{\fbox{\parbox{10cm}{Ce bloc [...] la ligne.}}}
\end{center}
```

2.2 L'idée consiste à créer une seconde ligne dans la `minipage` 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 9. 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.)

2.3 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 3

3.1 Les paquetages **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}
```

3.2 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}
```

3.3 Tout d'abord, remarquer 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{table}
```

3.4 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 notre document, il s'agit de placer la commande

```
\includepdf[pages=1]{formation_latex_UL-partie_2}
```

Chapitre 4

4.1 On trouve la commande pour produire le symbole ∘ dans le tableau 4.5. On peut produire les symboles de dérivée ′ avec la commande `\prime` (tableau 4.8) ou simplement avec le caractère `'`.

```
La dérivée de la fonction composée $f \circ g(x) =
f[g(x)]$ est $\{f[g(x)]\}^\prime = f^\prime[g(x)]
g^\prime(x)$. La dérivée seconde du produit des
fonctions $f$ et $g$ est
\begin{equation*}
  [f(x) g(x)]^{\prime\prime} =
  f^{\prime\prime}(x) g(x) + 2 f^\prime(x) g^\prime(x) +
  f(x) g^{\prime\prime}(x).
\end{equation*}
```

4.2 Nous avons aligné les lignes de l'équation juste à droite du premier symbole + à la première ligne. Remarquer l'usage des commandes `\cdots` et `\dots` dans la dernière ligne : LaTeX 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 +
    \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} \{\cdots\} + \dots +
    \frac{1}{n!} \{\cdots\} + R_n.
\end{align*}
```

4.3 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 5

5.1 On trouve les champs obligatoires et optionnels pour chaque type d'entrée dans Wikipedia. La clé est laissée vacante dans les réponses ci-dessous.

a) On utilise le type `Article` pour cette entrée.

```
@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},
  url = {https://www.tug.org/members/TUGboat/tb35-3/
         tb111mitt-float.pdf},
  language = {english}
}
```

b) On utilise le type `Book` pour cette entrée.

```
@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 =     {http://www.ctan.org/pkg/memoir/},
  language = {english}
}
```

c) La réponse ci-dessous contient les prénoms des auteurs, simplement afin d'illustrer que BibTeX sait les abréger au moment de composer la notice bibliographique. Remarquer l'utilisation des accolades { } 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 =    {http://www.ctan.org/pkg/pstricks-base/},
  language = {english}
}
```

Chapitre 6

6.1 a) `\newcommand{\NASA}{\textsc{nasa}}`

b) `\newcommand{\sigle}[1]{\textsc{\MakeLowercase{#1}}}`

c) `\newcommand{\sigle}[1]{\MakeUppercase{#1}}`

6.2 Avec la définition

```
\newcommand{\doc}[3][documentation]{%
  \href{#3}{#1~\raisebox{-0.2ex}{\faExternalLink}}%
  \marginpar{\faBook~\texttt{#2}}}
```

la déclaration à deux arguments

```
\doc{hyperref}{http://texdoc.net/pkg/hyperref}
```

hyperref produit toujours documentation.

6.3 La définition suivante donne les résultats demandés :

```
\newcommand{citation}
  {\begin{quote}
     \begin{tabular}{p{\linewidth}}
       \toprule\small\sffamily}%
  {\\ \bottomrule
     \end{tabular}
   \end{quote}}
```

On remarquera dans le troisième argument la présence de la commande de fin de ligne \\ qui doit absolument précéder `\bottomrule`.

Chapitre 7

7.1 On doit ajouter dans le préambule les commandes

```
\usepackage{mathpazo}
\usepackage{helvet}
```

La police Helvetica est très grande. Tel que mentionné dans la documentation de PSNFSS (section 4), il est généralement préférable de charger le paquetage **helvet** avec, par exemple, `psnfss2e`

```
\usepackage[scaled=0.92]{helvet}
```

pour que le texte en Helvetica se marie mieux à celui dans une autre police.

7.2 Tel qu'expliqué à la section 7.3, on doit insérer dans le préambule du document la commande

```
\hypersetup{linktocpage=true}
```

ou, plus simplement,

```
\hypersetup{linktocpage}
```

7.3 En conservant l'ajout de l'exercice précédent, on a dans le préambule la commande

```
\hypersetup{colorlinks, linktocpage}
```

Les hyperliens se présentent maintenant en couleur selon les paramètres par défaut de **hyperref**.

7.4 On peut soit utiliser des couleurs prédéfinies de **xcolor** (tableau 7.1), soit en définir de nouvelles avec `\definecolor`. On fait ensuite appel aux couleurs choisies pour 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}
```

7.5 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. URL https://tug.org/TUGboat/tb18-1/tb54becc.pdf.

Carlisle, D. 2014, *The **bm** Package*. URL http://www.ctan.org/pkg/bm/.

Carlisle, D. et The LaTeX3 Project. 2014, *Packages in the 'graphics' Bundle*. URL http://www.ctan.org/pkg/graphics/.

Chacon, S. et B. Straub. 2014, *Pro Git*, 2e éd., Apress. URL http://git-scm.com/book/fr/v2, version en ligne.

Collins-Sussman, B., B. W. Fitzpatrick et C. M. Pilato. 2008, *Gestion de versions avec Subversion*, O'Reilly Media. URL http://svnbook.red-bean.com, version en ligne.

Unicode Consortium, T. 2007, *The Unicode Standard, Version 5.0.0*, Addison-Wesley, Boston, ISBN 0-32148091-0.

Daly, P. W. 2010, *Natural Sciences Citations and References*. URL http://www.ctan.org/pkg/natbib/.

Danaux, X. et D. Gandy. 2015, *The **fontawesome** package. High quality web icons*. URL http://www.ctan.org/pkg/fontawesome/.

Fear, S. 2005, *Publication quality tables in LaTeX*. URL http://www.ctan.org/pkg/booktabs/.

Goulet, V. 2013, « Paquetage **francais-bst** », URL http://www.ctan.org/pkg/francais-bst/.

Goulet, V. 2015, *Rédaction de thèses et de mémoires avec LaTeX — Partie I*, Université Laval.

Heinz, C., B. Moses et J. Hoffmann. 2015, *The **listings** Package*. URL http://www.ctan.org/pkg/listings/.

Kern, D. U. 2007, *Extending LaTeX's color facilities: the **xcolor** package*. URL http://www.ctan.org/pkg/xcolor/.

Kline, P. et F. Charette. 2015, *biber — A Backend Bibliography Processor for Biblatex*. URL http://www.ctan.org/pkg/biber/.

Kopka, H. et P. W. Daly. 2003, *Guide to LaTeX*, 4e éd., Addison-Wesley, ISBN 978-0321173850.

Lehman, P. 2015, *The Biblatex Package — Programmable Bibliographies and Citations*. URL http://www.ctan.org/pkg/biblatex/.

Leisch, F. 2002, «Sweave: Dynamic generation of statistical reports using literate data analysis», dans *Compstat 2002 — Proceedings in Computational Statistics*, édité par W. Härdle et B. Rönz, Physica Verlag, Heidelberg, ISBN 3-7908-1517-9, p. 575–580. URL http://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. 2015, *The **pdfpages** Package*. URL http://www.ctan.org/pkg/pdfpages/.

Mittelbach, F. 2014, «How to influence the position of float environments like figure and table in LaTeX?», *TUGboat*, vol. 35, no 3, p. 258–254. URL https://www.tug.org/TUGboat/tb35-3/tb111mitt-float.pdf.

Mori, L. F. 2009, «Managing bibliographies with LaTeX», *TUGboat*, vol. 30, no 1, p. 36–48. URL https://www.tug.org/TUGboat/tb30-1/tb94mori.pdf.

Pakin, S. 2009, *The Comprehensive LaTeX Symbol List*. URL http://www.ctan.org/pkg/comprehensive/.

Patasnik, O. 1988, *BibTeXing*. URL http://www.ctan.org/pkg/bibtex/.

Rahtz, S. et H. Oberdiek. 2012, *Hypertext marks in LaTeX: a manual for **hyperref***. URL http://www.ctan.org/pkg/hyperref/.

Ramsey, N. 2011, «Noweb — a simple, extensible tool for literate programming», URL http://www.cs.tufts.edu/~nr/noweb/.

Robertson, W., P. Stephani et K. Hosny. 2014, *Experimental Unicode Mathematical Typesetting: The **unicode-math** Package*. URL http://www.ctan.org/pkg/unicode-math/.

American Mathematical Society. 2002, *User's Guide for the **amsmath** Package*. URL http://www.ctan.org/pkg/amsmath/.

Tantau, T. 2014, PGF *and TikZ — Graphic System for TEX*. URL http://www.ctan.org/pkg/pgf/.

Tantau, T., J. Wright et V. Miletić. 2015, *The* BEAMER *Class*. URL http://www.ctan.org/pkg/beamer/.

Wikilivres. 2015, « LaTeX », URL http://fr.wikibooks.org/wiki/LaTeX.

Wikipedia. 2015, «Mathematical operators and symbols in Unicode», URL https://en.wikipedia.org/wiki/Mathematical_operators_and_symbols_in_Unicode.

Wilson, P. 2013, *The Memoir Class for Configurable Typesetting*, 8e éd., The Herries Press. URL http://www.ctan.org/pkg/memoir/, maintained by Lars Madsen.

Xie, Y. 2016, «**knitr**: A general-purpose package for dynamic report generation in R», URL http://cran.r-project.org/package=knitr, R package version 1.12.

Van Zandt, T., D. Girou et H. Voß. 2014, *PSTricks — PostScript Macros for Generic TEX*. URL http://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.

\,, 41, 43
\:, 43
\;, 43

align (environnement), 50–53, 55–59, 104
aligned (environnement), 54
\alpha, 46, 48
amsfonts (paquetage), 46, 61
amsmath (paquetage), 36, 37, 39–42, 44, 47–52, 54, 55, 58, 111
\angle, 25, 31, 49
\approx, 56
\aption, 27
\arbox, 10
\arccos, 44
\arcsin, 44
\arctan, 44
\arg, 44
\ast, 49
\autoref, 29

babel (paquetage), 9, 29, 31, 68, 74, 78, 102
\backslash, 49
\bar, 49
beamer (classe), 96, 99
\beta, 46
biblatex (paquetage), 64
\bibliography, 73, 75
\bibliographystyle, 69, 73, 75
\Big, 55
\big, 55
\Bigg, 55
\bigg, 55
\bm, 57, 58
bm (paquetage), 57, 109
Bmatrix (environnement), 61
bmatrix (environnement), 59, 61
booktabs (paquetage), 16, 17, 19
\bottomrule, 19, 20, 28, 30, 102, 103, 107
\boxplus, 49
\boxtimes, 49

\cap, 49
\caption, 29, 30, 33, 103
cases (environnement), 54, 60, 105

\cdot, 49
\cdots, 41, 42, 104
center (environnement), 101
\centering, 22, 28, 30
\chaptitlefont, 80
\chi, 46
\circ, 49, 104
\circle, 22, 23
\citation, 83, 91, 107
citation (environnement), 83, 85
\cite, 71
\citeauthor, 72
\citep, 71, 75
\citet, 71, 73, 75
\citeyear, 72
classe
  beamer, 96, 99
  memoir, v, 17, 19, 21, 28, 31, 32, 80, 81, 102, 120
  slides, 96
  ulthese, v, vii, 2, 17, 24, 32, 64, 71, 78, 91, 95, 97
\cmidrule, 19
\color, 89, 93, 108
\colorlet, 83, 89
\cos, 44, 94, 95
\cosh, 44
\cot, 44
\coth, 44
\csc, 44
\cup, 49

\dashbox, 22
\ddot, 49
\ddots, 41
\DeclareMathOperator, 44
\definecolor, 89–91, 93, 108
\deg, 44
\Delta, 46
\delta, 46
\det, 44
\dfrac, 39, 40, 59–61, 104, 105
\dim, 44
\div, 49
\dots, 41, 42, 104
\dotsb, 42
\dotsc, 42
\dotsi, 42
\dotsm, 42
\dotso, 42
\Downarrow, 49
\downarrow, 49

\ell, 49
\emptyset, 49
environnement
  align, 50–53, 55–59, 104
  aligned, 54
  Bmatrix, 61
  bmatrix, 59, 61
  cases, 54, 60, 105
  center, 101
  citation, 83, 85
  equation, 36, 39, 44, 45, 48, 51, 104, 105
  figure, 27, 31, 33
  gather, 51
  lstlisting, 92, 108
  minipage, 10, 11, 13, 22, 31, 84, 101–103
  multline, 51
  NoHyper, 73
  picture, 21–23
  pmatrix, 61
  quote, 83, 107
  Sinput, 94
  Soutput, 94
  split, 50–52
  table, 27, 28, 30, 33, 83, 84, 103

tabular, 17–20, 28, 30, 102, 103, 107
tabular*, 17, 18
tabularx, 17, 18, 20, 102, 103
titled-frame, 84
verbatim, 91, 92
Vmatrix, 61
vmatrix, 61
\epsilon, 46
\eqref, 52
equation (environnement), 36, 39, 44, 45, 48, 51, 104, 105
\equiv, 49
\eta, 46
\exists, 49
\exp, 44, 94, 95

\fbox, 8, 101
figure (environnement), 27, 31, 33
fontawesome (paquetage), 82, 109
fontenc (paquetage), 97
fontspec (paquetage), 69, 74, 87
\forall, 49
\frac, 39, 40, 45, 48, 53–56, 59, 61, 104
\frame, 83, 84, 93
\framebox, 8, 9, 11, 22
framed (paquetage), 84
francais-bst (paquetage), 70, 78, 109
\frenchfigurename, 29
\frenchtablename, 29

\Gamma, 46, 48, 57, 58
\gamma, 46
gather (environnement), 51
\gcd, 44
\geq, 49, 54
\gets, 49
graphics (paquetage), 23
graphicx (paquetage), 4, 23, 25

\hat, 49
\height, 24
helvet (paquetage), 98, 107
\hfill, 11
\hom, 44
\href, 82
hyperref (paquetage), 73, 82, 90, 91, 98, 107, 108, 110
\hypersetup, 90, 91, 107, 108

\ieme, 9
\iiint, 40
\iint, 40, 41
\in, 49
\include, 2–4
\includegraphics, 4, 5, 23–25, 31
\includeonly, 3–5
\includepdf, 26, 103
\inf, 44
\infty, 44, 45, 48, 49
\input, 1–3, 5
inputenc (paquetage), 69, 97
\int, 40, 44, 45, 48, 54, 56
\iota, 46

\kappa, 46
\ker, 44
knitr (paquetage), 111

\label, 29, 30, 51
\Lambda, 46
\lambda, 46, 48
\language, 93
\ldots, 41
\le, 11
\left, 53–56, 59, 104
\Leftarrow, 49
\leftarrow, 49
\Leftrightarrow, 49
\leftrightarrow, 49
\leq, 49, 105

\lg, 44
\lim, 44
\liminf, 44
\limsup, 44
\line, 23
\linewidth, 8, 11, 31, 84, 103, 107
listings (paquetage), 92, 93, 98, 110
\listoffigures, 29
\listoftables, 29
\ln, 44
\log, 43, 44
\Longleftarrow, 49
\longleftarrow, 49
\Longleftrightarrow, 49
\longleftrightarrow, 49
\Longrightarrow, 49
\longrightarrow, 49
lstlisting (environnement), 92, 108

\makebox, 8, 9, 22
\MakeLowercase, 85, 106
\MakeUppercase, 106
\marginpar, 82, 106
\mathbb, 46, 47, 80
\mathbf, 57
\mathcal, 46
mathpazo (paquetage), 88
mathptmx (paquetage), 88
\max, 44, 52
\mbox, 8, 9, 13, 101, 102
memoir (classe), v, 17, 19, 21, 28, 31, 32, 80, 81, 102, 120
\midrule, 19, 20, 28, 30, 102, 103
\min, 44
minipage (environnement), 10, 11, 13, 22, 31, 84, 101–103
\mu, 46
\multicolumn, 20, 102
multline (environnement), 51

\nabla, 49
natbib (paquetage), 63–65, 68–72, 74, 78
\neg, 49
\neq, 49
\newcommand, 58, 73, 80–82, 85, 106, 107
\newenvironment, 82, 83
\newline, 31
\newsubfloat, 33, 103
\ngeq, 49
\ngtr, 49
\nLeftarrow, 49
\nleftarrow, 49
\nLeftrightarrow, 49
\nleftrightarrow, 49
\nleq, 49
\nless, 49
\No, 31
\nocite, 72
NoHyper (environnement), 73
\noindent, 84
\nombre, 31
\normalfont, 81
\not, 47
\notag, 50
\notin, 49
\nRightarrow, 49
\nrightarrow, 49
\nu, 46
numprint (paquetage), 31, 102

\Omega, 46
\omega, 46
\oplus, 49
\otimes, 49

paquetage
    amsfonts, 46, 61

amsmath, 36, 37, 39–42, 44, 47–52, 54, 55, 58, 111
babel, 9, 29, 31, 68, 74, 78, 102
biblatex, 64
bm, 57, 109
booktabs, 16, 17, 19
fontawesome, 82, 109
fontenc, 97
fontspec, 69, 74, 87
framed, 84
francais-bst, 70, 78, 109
graphics, 23
graphicx, 4, 23, 25
helvet, 98, 107
hyperref, 73, 82, 90, 91, 98, 107, 108, 110
inputenc, 69, 97
knitr, 111
listings, 92, 93, 98, 110
mathpazo, 88
mathptmx, 88
natbib, 63–65, 68–72, 74, 78
numprint, 31, 102
pdfpages, 26, 33, 103, 110
tabularx, 17
unicode-math, 46, 58, 59, 111
xcolor, 88–90, 93, 98, 108, 110
\parbox, 10, 11, 13, 101, 102
\partial, 49, 59, 104
pdfpages (paquetage), 26, 33, 103, 110
\phantom, 50, 51, 55, 56
\Phi, 46
\phi, 46
\Pi, 46
\pi, 46, 94, 95
picture (environnement), 21–23
\pm, 49
pmatrix (environnement), 61
\Pr, 44
\prime, 49, 104
\Psi, 46
\psi, 46
\put, 22, 23

\qbezier, 23
\qquad, 42, 43
\quad, 42, 43, 48, 54, 59
quote (environnement), 83, 107

\raisebox, 9, 82, 84, 106
\ref, 29
\reflectbox, 25, 26, 31
\renewcommand, 80, 81, 85
\renewenvironment, 82
\resizebox, 25, 26
\rho, 46
\right, 53–57, 59, 104
\Rightarrow, 49
\rightarrow, 49
\rotatebox, 25, 26
\rule, 12, 20

\scalebox, 25, 26
\sec, 44
\section, 94
\setmainfont, 88, 99
\setmathfont, 88, 99
\setminus, 49
\setmonofont, 99
\setsansfont, 99
\Sexpr, 94
\Sigma, 46
\sigma, 46
\sin, 43, 44
\sinh, 44
Sinput (environnement), 94
slides (classe), 96
Soutput (environnement), 94
split (environnement), 50–52
\sqrt, 40

\star, 49
\subcaption, 33, 103
\subset, 49
\subseteq, 49
\sum, 40, 48, 53, 56
\sup, 44
\symbf, 58

table (environnement), 27, 28, 30, 33, 83, 84, 103
tabular (environnement), 17–20, 28, 30, 102, 103, 107
tabular* (environnement), 17, 18
tabularx (environnement), 17, 18, 20, 102, 103
tabularx (paquetage), 17
\tan, 44
\tanh, 44
\tau, 46
\text, 42, 48, 52
\textbf, 57
\textsc, 85, 106
\textsubscript, 9
\textsuperscript, 9
\textwidth, 5, 8, 11, 18, 20, 102
\tfrac, 39, 40, 61
\Theta, 46
\theta, 46, 57–59
\thicklines, 22, 23
\thinlines, 23
\tilde, 49, 105
\times, 49, 59
titled-frame (environnement), 84
\to, 44, 49
\toprule, 19, 20, 28, 30, 102, 107

ulthese (classe), v, vii, 2, 17, 24, 32, 64, 71, 78, 91, 95, 97
unicode-math (paquetage), 46, 58, 59, 111
\unitlength, 22
\Uparrow, 49
\uparrow, 49
\Updownarrow, 49
\updownarrow, 49
\Upsilon, 46
\upsilon, 46

\varepsilon, 46
\varphi, 46
\varpi, 46
\varrho, 46
\varsigma, 46
\vartheta, 46
\vdots, 41
\vector, 23
\vee, 49
\verb, 94, 95
verbatim (environnement), 91, 92
Vmatrix (environnement), 61
vmatrix (environnement), 61

\wedge, 49
\width, 5

xcolor (paquetage), 88–90, 93, 98, 108, 110
\Xi, 46
\xi, 46

\zeta, 46

Ce document a été produit avec le système de mise en page XeLaTeX 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 Adobe Myriad Pro. Les icônes proviennent de la police Font Awesome.

ISBN 978-2-9811416-5-1