

Rédaction de thèses et de mémoires avec **L^AT_EX**

2. Concepts avancés

14 septembre 2015

© 2015 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, professeur titulaire, avec la contribution financière de la Bibliothèque de l'Université Laval.


Introduction


Ce document constitue la seconde partie d'une formation sur l'utilisation de \LaTeX pour la rédaction de thèses et de mémoires développée pour la Bibliothèque de l'Université Laval. La première partie de la formation se déroulant en classe, sa documentation n'existe que sous forme de diapositives (Goulet, 2015).


Ce document reprend 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, etc.

Le présent manuel 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 (plus de 600 pages !) documentation de la classe memoir (Wilson, 2013), sur laquelle se base la classe ulthese, 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 *T_EX- \LaTeX Stack Exchange*  (avant de poser une question, vérifiez que la réponse ne se trouve pas déjà dans ce 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.

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

 ulthese

qu'illustré ici : un hyperlien mène vers la version en ligne de la documentation, telle qu'on la trouve dans la distribution T_EX Live, dans le site [TeXdoc Online](#) . De plus, on indique dans la marge le nom du fichier (sans l'extension .pdf) correspondant sur un système doté de T_EX Live. Sur la plupart des systèmes, il est possible de consulter hors ligne la documentation *fichier.pdf* en entrant à une invite de commande

```
texdoc fichier
```

Chaque chapitre comporte quelques exercices. Les solutions sont fournies en annexe. Le cas échéant, le numéro d'un exercice est un hyperlien vers sa solution, et vice versa. Le document est accompagné de fichiers pour certains exercices. Pour les autres, utiliser le fichier `exercice_gabarit.tex` comme gabarit pour composer des solutions.

Table des matières

Introduction [iii](#)

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	Boîtes horizontales	7
2.2	Boîtes verticales	9
2.3	Boîtes de réglure	11
2.4	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	44
4.5	Équations sur plusieurs lignes et numérotation	47
4.6	Délimiteurs de taille variable	52
4.7	Caractères gras en mathématiques	56
4.8	Exercices	58

A Solutions des exercices	35
Chapitre 2	35
Chapitre 3	36
Bibliographie	39

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
```



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

\end{document}
```

**Conseil du T_EXpert**

Utiliser 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`.

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**.

- 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.
- 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 `preamble.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 pour disposer de l'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'utilisateur.

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

2.1 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 droite¹. Le contenu est normalement du texte, mais conceptuellement ce pourrait être n'importe quoi, y compris d'autres boîtes.

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

Les commandes de base pour créer des boîtes horizontales sont :

```
\mbox{texte}
\fbbox{texte}
```

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



Conseil du T_EXpert

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 T_EX 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](#).

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

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

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

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

produit aligné à gauche, alors que

```
\makebox[3.5cm]{centré}
```

produit centré.

Il arrive que l'on doive ajuster légèrement le positionnement vertical d'éléments de contenu, notamment pour les images. La commande

```
\raisebox{déplacement}{texte}
```


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

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

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

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

produit _{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`, le mode mathématique ([chapitre 4](#)) ou les commandes de la famille `\i` de **babel** (section 1.1 de la [documentation](#) .

 frenchb

2.2 Boîtes verticales

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

La commande de base pour créer une boîte verticale est :

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

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

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

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

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

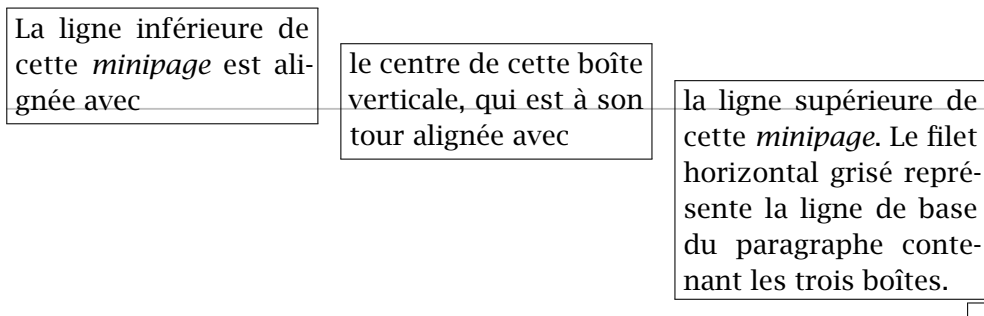
La signification des arguments *largeur* et *pos* est la même que pour la commande `parbox`.

L'environnement `minipage` est souvent 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

```
\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 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[\textwidth]{gauche \hfill droite}
```

produit



alors que

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

produit

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

2.3 Boîtes de réglure

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

La commande

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



crée une réglure de dimensions *largeur* \times *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 *déplacement* est positif ou négatif. 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 paraît 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](#).

3. *Rule box*, en anglais

2.4 Exercices

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

- 2.1** Une fois qu'une boîte est définie, $\text{T}_{\text{E}}\text{X}$ n'y voit qu'une unité de contenu avec ses dimensions propres. Il est donc possible de définir une boîte à l'intérieur d'une autre, et ce, peu importe le type de boîte.

Avec ceci en tête, définir la boîte suivante :

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

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

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

- 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 filets¹ 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** 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 :

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 arguments² 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 !\[\]\(3cb60d42b10e53f9522bb0b392c1c4cd_img.jpg\)](#)).

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 contenu du 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, pour définir un tableau à trois colonnes dont le contenu de la première est aligné à gauche, celui de la seconde à droite et celui de la troisième en texte libre dans une cellule de 5 cm de largeur, on utiliserait :

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

Avec la déclaration suivante, la largeur de la troisième colonne sera automatiquement adaptée pour que le tableau occupe toute la largeur de la page :

```
\begin{tabularx}{\textwidth}{lrx}
```

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 autrement alignée à gauche ou à droite.

Les commandes suivantes³ 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{n-m}` insère un filet horizontal comme `\midrule`, mais seulement de la gauche de la colonne *n* à la droite de la colonne *m* ;

`\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`.

Un exemple simple de lignes de contenu serait :

```
\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
```

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

3. Ce sont les commandes de **booktabs** et memoir auxquelles nous faisons référence à la [section 3.1](#).

Exemple 3.1. On reprend le contenu ci-dessus pour en faire un tableau d’une largeur ajustée automatiquement au contenu. La première colonne est alignée à gauche et toutes les autres à droite.

```
\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}
```

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 occupe maintenant 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 2.3](#)).

```
\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 intéressé d'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 la même que celle du document.

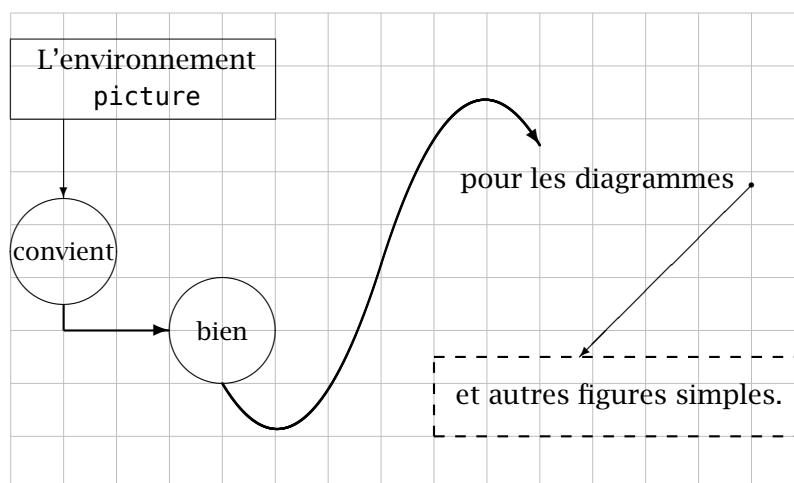
Pour tracer une figure avec l'environnement `picture`, on crée d'abord une grille (invisible) d'une dimension quelconque dans l'unité de mesure de son choix (autrement dit : les lignes de la grille peuvent être distantes aussi

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 quelque'autre information pour compléter l'élément. C'est souvent plus simple d'esquisser d'abord un modèle au crayon sur du papier quadrillé.

 memman

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 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 de 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 de 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 est 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
\qbezier(4,2)(5.5,-0.5)(7,4.25) % deux courbes de Bézier bout à
\qbezier(7,4.25)(8.5,9)(10,6.5) % bout pour produire courbe en S
\put(10,6.5){\vector(2,-3){0}} % pointe de flèche seule
\end{picture}

```

Il existe quelques outils pour tracer des figures plus complexes directement avec \TeX , dont PSTricks (Van Zandt et collab., 2014) ou le système TikZ/PGF (Tantau, 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 paquets **graphics** ou


graphicx (Carlisle et The L^AT_EX3 Project, 2014) en combinaison avec un moteur T_EX moderne tel que pdfL^AT_EX ou X_YL^AT_EX. Les fonctionnalités des deux paquets 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 L^AT_EX 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 pdfL^AT_EX et X_YL^AT_EX, 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}
```



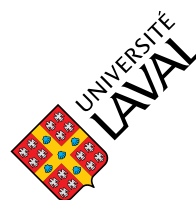
```
%% réduction à 15 mm de haut
\includegraphics[height=15mm]{ul_p}
```



(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}
```


9tX9t


□

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

Pourquoi 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 remplie⁵. 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 :

4. *Float* en anglais.


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

- 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.

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. □

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

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

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

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

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

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



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 — ajouter dans le préambule les commandes suivantes :

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

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}{lrrrr}
    \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


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.

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](#). \square

Les environnements `table` et `figure` créent des éléments flottants qui,

par ailleurs, sont des boîtes verticales standards (section 2.2). Il est donc permis d'y mettre à peu près n'importe quoi, mais surtout plus d'un tableau ou plus d'une figure (ou même une combinaison des deux). Les environnements `minipage` (section 2.2) se révèlent 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`.

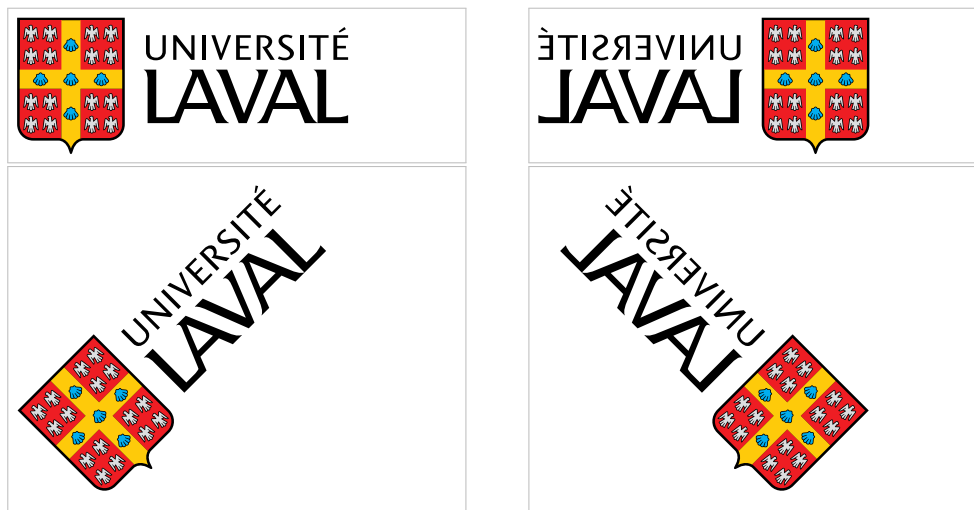


FIG. 3.2 – Exemple de disposition de plusieurs graphiques dans une même figure flottante

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

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

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 intention d'en créer pour les environnements flottants `table` ou `figure` avec, selon le cas, les commandes

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

Ensuite, on utilise la commande

```
\subcaption{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 manières d'activer le mode mathématique :

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

on sait que $\$(a + b)^2 = a^2 + 2ab + b^2\$, d'o\grave{u}$
on obtient...

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

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

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

on sait que

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

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

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

on sait que

$$(a + b)^2 = a^2 + b^2, \quad (4.1)$$
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 (??? capsule vidéo ???); nous y reviendrons. 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 le rendu. 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 multi-lignes ;
- ▶ une meilleure gestion de l'espacement autour des signes d'égalité dans les équations multi-lignes ;
- ▶ 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 moins avancé devrait impérativement consulter la [documentation complète](#)  du paquetage.

4.3 Principaux éléments du mode mathématique

4.3.1 Exposants et indices

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

- On place un caractère en exposant d'un autre avec la commande `^` et en indice avec la commande `_`. Les indices et exposants se combinent naturellement.

$$x^2 \quad a_n \quad x_i^k$$

(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 `{ }`.

$$x^{2k+1} \quad x_{i,j} \quad x_{ij}^{2n} \quad x_{ij}^{2n}$$

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

$$e^{-x^2} \quad A_{i_s, k^n}^{y_i}$$



Les commandes `^` et `_` sont permises dans le mode mathématique seulement. En fait, si ℒ_T 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}}{y} + 1  
\end{equation*}
```

Soit

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

- Les commandes

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

de **amsmath** permettent de forcer une fraction à adopter la taille d'une fraction hors paragraphe (*displayed*) dans le cas de `\dfrac` ou de celle d'une fraction au fil du texte (*text*) dans le cas de `\tfrac`.

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

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

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

TAB. 4.1 – Points de suspension

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

- 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 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. Il faut éviter de les entrer comme trois points finaux consécutifs, car l'espacement entre ceux-ci sera trop petit et le résultat, disgracieux : ...

- 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.

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

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

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

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

- Le paquetage **amsmath** 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
```

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

- La commande `\quad` insère un blanc de 1 em, soit une longueur égale à la taille de la police de caractère courante en points (donc 11 points pour une police de 11 points). La commande `\qqquad` insère le double de cette longueur.¹

```
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.

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

TAB. 4.2 – Espaces dans le mode mathématique

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

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 :

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

Utiliser à répétition la commande `text` se révèle peu pratique à l'usage.

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

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

L'espacement autour des opérateurs mathématiques est également géré par \LaTeX .

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

- 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
```

$$\lim_{x \rightarrow 0} x = 0$$

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

$$\lim_{x \rightarrow 0} x = 0$$

 `amslatex`


- La commande `\DeclareMathOperator` de **amsmath** permet de définir de nouveaux noms de fonctions ; consulter la [documentation](#)  du paquetage (section 5.1) pour les détails.

4.4 Symboles mathématiques

Outre les chiffres et les lettres de l'alphabet, les claviers d'ordinateurs comptent somme toute très peu des innombrables symboles mathématiques utilisés couramment. Pour les représenter dans \LaTeX , 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.

 `comprehensive`

L'ouvrage de référence pour connaître les symboles disponibles dans \LaTeX est la bien nommée [Comprehensive \$\text{\LaTeX}\$ Symbol List](#)  (Pakin, 2009). Dans l'édition de 2009, la liste comprenait pas moins de 5 913 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.



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

TAB. 4.4 - Lettres grecques majuscules

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

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 .

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*) des lettres de l'alphabet. Elles sont principalement utilisées pour représenter les ensembles de nombres.

<code>\mathbb{NZRC}</code>	\mathbb{NZRC}
----------------------------	-----------------

- Le tableau 213 de la [Comprehensive \$\text{\LaTeX}\$ Symbol List](#)  présente plusieurs autres alphabets spéciaux disponibles en mode mathématique.
- 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 : $\mathcal{ABC}xyz$. Consulter éventuellement la documentation de la police pour les détails.


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 \$\text{\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 48](#) ne présentent que les principaux éléments à titre indicatif.
- Certaines relations existent directement en version opposée, ou négative (comme \neq ou \nsubseteq) 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 \$\text{\LaTeX}\$ Symbol List](#) . Le document contient divers autres types de flèches, mais celles-ci ne sont généralement pas appropriées pour les mathématiques (pensons à $\wedge\rightarrow$ 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 Autres symboles utiles

Le [tableau 4.8](#) présente quelques autres symboles mathématiques plus fréquemment utilisés fournis par \LaTeX .

Exemple 4.1. Les éléments de cette section et de la précédente permettent déjà de créer des équations élaborées. Considérons l'équation suivante :

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

Voici, pièce par pièce, le code \LaTeX pour créer cette équation.

<code>\begin{equation*}</code>	<i>équation hors paragraphe</i>
<code>\frac{\Gamma(\alpha)}{\lambda^\alpha} =</code>	$\frac{\Gamma(\alpha)}{\lambda^\alpha} =$
<code>\sum_{j = 0}^{\infty} \int_j^{j + 1}</code>	$\sum_{j=0}^{\infty} \int_j^{j+1}$
<code>x^{\alpha - 1} e^{-\lambda x} \backslash, dx</code>	$x^{\alpha-1} e^{-\lambda x} dx$
<code>, \quad \alpha > 0 \text{ et } \lambda > 0.</code>	$, \quad \alpha > 0 \text{ et } \lambda > 0.$
<code>\end{equation*}</code>	<i>fin de l'environnement</i>

□

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

TAB. 4.5 - Quelques opérateurs binaires

\times	<code>\times</code>	\div	<code>\div</code>	\pm	<code>\pm</code>	\mp	<code>\mp</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\setminus	<code>\setminus</code>	\cdot	<code>\cdot</code>
\wedge	<code>\wedge</code>	\vee	<code>\vee</code>	\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>

TAB. 4.6 - Quelques relations et leur négation

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

[†] requiert **amsmath**

TAB. 4.7 - Quelques flèches et leur négation

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

[†] requiert **amsmath**

TAB. 4.8 - Symboles mathématiques divers

∞	<code>\infty</code>	∇	<code>\nabla</code>	∂	<code>\partial</code>	ℓ	<code>\ell</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\emptyset	<code>\emptyset</code>	$'$	<code>\prime</code>
\neg	<code>\neg</code>	\backslash	<code>\backslash</code>	$\ $	<code>\ </code>	\angle	<code>\angle</code>

manière de représenter des groupes d'équations du type

$$y = 2x + 4 \quad (4.2)$$

$$y = 6x - 1 \quad (4.3)$$

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`.
- 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 `\\`.

<pre>\begin{align} a_1 &=& b_1 + c_1 \notag \\ a_2 &=& b_2 + c_2 - d_2 + e_2 \\ \end{align}</pre>	$\begin{aligned} a_1 &= b_1 + c_1 \\ a_2 &= b_2 + c_2 - d_2 + e_2 \end{aligned} \quad (4.13)$
---	---

- Les renvois vers des équations numérotées fonctionnent, comme partout ailleurs en \LaTeX , avec le système d'étiquettes et de références ([Goulet, 2015](#), ou la [section 3.4](#)). Le paquetage **amsmath** fournit également la pratique commande `\eqref` qui place automatiquement le numéro d'équation entre parenthèses.

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

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

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.

Consulter le chapitre 3 de la [documentation](#) du paquetage **amsmath** pour les détails sur l'utilisation des environnements du tableau 4.9.

 **amsl doc**



Conseil du TeXpert

Veillez à respecter les règles suivantes pour la mise en forme des équations :

1. Qu'elles apparaissent en ligne ou hors paragraphe, les équations font partie intégrante de la phrase, aussi 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 numéroté que les équations d'un document auxquelles le texte fait référence.

Exemple 4.2. Les deux suites d'équations au début de la section ont été réalisées avec les blocs de code suivants, 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 de se retrouver, par exemple, non pas avec

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

mais plutôt avec

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

- La paire de commandes

```
\leftdelim_g ... \rightdelim_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.

<pre>\left(1 + \frac{1}{x} \right)</pre>	$\left(1 + \frac{1}{x}\right)$
<pre>\left(\sum_{i = 1}^n x_i^2 \right)^{1/2}</pre>	$\left(\sum_{i=1}^n x_i^2\right)^{1/2}$

- 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`.

<pre>\left[\int \left(1 + \frac{x}{k} \right) dx \right]</pre>	$\left[\int \left(1 + \frac{x}{k}\right) dx\right]$
--	---

- 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.
```

$$f(x) = \begin{cases} 1 - x, & x < 1 \\ x - 1, & x \geq 1 \end{cases}$$

(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 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.

TAB. 4.10 - Tailles des délimiteurs mathématiques

taille standard	() [] { }
<code>\big</code>	() [] { }
<code>\Big</code>	() [] { }
<code>\bigg</code>	() [] { }
<code>\Bigg</code>	() [] { }

```
\begin{align*}
a
&= \left(
b + \frac{c}{d}
\right. \\
&\quad \left. + \frac{e}{f} - g
\right)
\end{align*}
```

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

- 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.

 `amslatex`

- À lire également : la section 14 de la [documentation](#) de **amsmath** qui traite de divers enjeux typographiques en lien avec les délimiteurs et qui introduit des nouvelles commandes pour contrôler leur taille.

Exemple 4.3. Le développement de la formule d'approximation de Simpson comporte plusieurs des éléments discutés jusqu'à maintenant :

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

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

Le code source de cette suite d'équations est le suivant :

```

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

```

□



Conseil du T_EXpert

De manière générale, il est déconseillé de scinder une suite d'équations entre deux pages. Le chargement du paquetage **amsmath** désactive d'ailleurs cette possibilité. Cependant, c'est parfois inévitable pour les longs blocs d'équations.

La commande `\displaybreak`, placée immédiatement avant `\` permet d'indiquer à L^AT_EX la possibilité d'insérer un saut de page après la ligne courante dans le bloc d'équations.

La commande accepte en argument optionnel un entier entre 0 et 4 indiquant à quel point un saut de page est désiré : `\displaybreak[0]` signifie «il est permis de changer de page ici» sans que ce ne soit obligatoire ; `\displaybreak[4]` ou, de manière équivalente, `\displaybreak` force un saut de page.

À moins d'en être vraiment aux toutes dernières étapes d'édition d'un document, il est conseillé d'utiliser la commande `\displaybreak` avec parcimonie et avec un argument optionnel faible.

4.7 Caractères gras en mathématiques

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

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

Pourquoi consacrer une section spécialement à cette convention typographique ? Parce que la création de symboles mathématiques en gras doit certainement figurer parmi les questions les plus fréquemment posées par les utilisateurs de L^AT_EX... 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 L^AT_EX 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}
```

$$\theta\theta + \Gamma\Gamma + AA$$

On remarquera également que `\mathbf{A}` produit une lettre majuscule droite plutôt qu'italique.

- La manière la plus standard et robuste d'obtenir des symboles mathématiques 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}
```

$$\theta\theta + \Gamma\Gamma + AA$$

- Les utilisateurs de \LaTeX 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`

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

$$\theta\theta + \Gamma\Gamma + AA$$

- Si le gras est fréquemment utilisé dans un document pour une notation particulière, il est fortement recommandé de définir une nouvelle commande² 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

2. La définition de nouvelles commandes est couvert plus en détail à la ??.

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

Exemple 4.4. 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` ci-dessus est définie dans le préambule du document. On obtient l'équation

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

avec le code source suivant :

```
\begin{align*}
\mat{J}(\mat{\theta})
&=
\begin{bmatrix}
\frac{\partial f_1(\mat{\theta})}{\partial \theta_1} & \frac{\partial f_1(\mat{\theta})}{\partial \theta_2} \\
\frac{\partial f_2(\mat{\theta})}{\partial \theta_1} & \frac{\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*}
```

□

4.8 Exercices

Bibliographie

- Carlisle, D. 2014, *The **bm** Package*. URL <http://www.ctan.org/pkg/bm/>.
- Carlisle, D. et The L^AT_EX₃ Project. 2014, *Packages in the ‘graphics’ Bundle*. URL <http://www.ctan.org/pkg/graphics/>.
- Unicode Consortium, T. 2007, *The Unicode Standard, Version 5.0.0*, Addison-Wesley, Boston, ISBN 0-32148091-0.
- Goulet, V. 2015, *Rédaction de thèses et de mémoires avec L^AT_EX — Partie I*, Université Laval.
- Kopka, H. et P. W. Daly. 2003, *Guide to L^AT_EX*, 4^e éd., Addison-Wesley, ISBN 978-0321173850.
- 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 L^AT_EX?», *TUGboat*, vol. 35, n° 3, p. 258-254.
- Pakin, S. 2009, *The Comprehensive L^AT_EX Symbol List*. URL <http://www.ctan.org/pkg/comprehensive/>.
- 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 T_EX*. URL <http://www.ctan.org/pkg/pgf/>.
- Wikilivres. 2015, «L^AT_EX», 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*, 8^e éd., The Herries Press. URL <http://www.ctan.org/pkg/memoir/>, maintained by Lars Madsen.

Van Zandt, T., D. Girou et H. Voß. 2014, *PSTricks — PostScript Macros for Generic T_EX*. URL <http://www.ctan.org/pkg/pstricks-base/>.

Ce document a été produit avec le système de mise en page Xe_{La}T_EX à partir de la classe memoir. Le texte principal est en Lucida Bright OT 11 points, le code informatique en Bitstream Vera Sans Mono et les titres en Adobe Myriad Pro. Les icônes proviennent de la police Font Awesome.

