Une courte (?) introduction à LaTeX 2ε

ou LaTeX2e en 84 minutes

par Tobias Oetiker

Hubert Partl, Irene Hyna et Elisabeth Schlegl

traduit en français par Matthieu Herrb

Version 3.20
Novembre 2001

Copyright © 1998-2001 Tobias Oetiker et tous les contributeurs de LShort.
Copyright © 1998-2001 LAAS/CNRS pour la traduction.

Ce document est libre ; vous pouvez le redistribuer et/ou le modifier selon les termes de la licence publique générale de GNU publiée par la Free Software Foundation (version 2 ou tout autre version ultérieure choisie par vous)

Ce document est diffusé en espérant qu'il sera utile, mais SANS AUCUNE GARANTIE, ni explicite ni implicite, y compris les garanties de commercialisation ou d'adaptation dans un but spécifique. Reportez-vous à la licence publique générale de GNU pour plus de détails.

Vous devez avoir reçu une copie de la licence publique générale de GNU en même temps que ce document. Si ce n'est pas le cas, écrivez à la Free Software Fundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, États-Unis.

Copyright © 1998 Tobias Oetiker and all the Contributers to LShort. All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.

Merci !

Ce document est une traduction en français de « *The not so short introduction to LaTeX2e* » par Tobias Oetiker.
Une grande partie de ce document provient d'une introduction autrichienne à LaTeX 2.09, écrite en allemand par :

Hubert Partl `<partl@mail.boku.ac.at>`
Zentraler Informatikdienst der Universität für Bodenkultur, Wien

Irene Hyna `<Irene.Hyna@bmwf.ac.at>`
Bundesministerium für Wissenschaft und Forschung, Wien

Elisabeth Schlegl `<no email>`
in Graz

La version courante en français est disponible sur :
`CTAN:/info/lshort/french/`[1]

Vous trouverez la version anglaise de Tobias Oetiker sur :
`CTAN:/info/lshort/english/`

Si vous êtes intéressés par la version allemande, vous trouverez une version adaptée à LaTeX 2_ε par Jörg Knappen sur :
`CTAN:/info/lshort/german/`

[1] Voir page vi la liste des sites `CTAN`.

Pour la préparation de ce document, l'aide des lecteurs du forum Usenet `comp.text.tex` a été sollicitée. De nombreuses personnes ont répondu et ont fourni des corrections, des suggestions et du texte pour améliorer ce document. Qu'ils en soient ici remerciés sincèrement. Ajoutons que je suis responsable de toutes les erreurs que vous pourriez trouver dans ce document.

Merci en particulier à :

Rosemary Bailey, David Carlisle, Christopher Chin, Chris McCormack, Wim van Dam, David Dureisseix, Elliot, David Frey, Robin Fairbairns, Alexandre Guimond, Cyril Goutte, Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehradt, Martin Maechler, Claus Malten, Hubert Partl, John Refling, Mike Ressler, Brian Ripley, Young U. Ryu, Chris Rowley, Hanspeter Schmid, Craig Schlenter, Josef Tkadlec, Didier Verna, Fabian Wernli, Fritz Zaucker, Rick Zaccone et Mikhail Zotov

La version française a bénéficié des contributions des lecteurs du forum `fr.comp.text.tex` et en particulier de :

Sebastien Blondeel, Marie-Dominique Cabanne, Christophe Dousson, Olivier Dupuis, Daniel Flipo, Paul Gaborit, Thomas Ribo, Philippe Spiesser et Vincent Zoonekynd.

Note du traducteur : je tiens également à remercier chaleureusement les auteurs de ce document de le rendre publiquement utilisable et d'avoir ainsi rendu possible cette version française.

Préface

LaTeX[1] est un logiciel de composition typographique adapté à la production de documents scientifiques et mathématiques de grande qualité typographique. Il permet également de produire toutes sortes d'autres documents, qu'il s'agisse de simples lettres ou de livres entiers. LaTeX utilise TeX[2] comme outil de mise en page.

Cette introduction décrit LaTeX 2_ε et devrait se montrer suffisante pour la plupart des applications de LaTeX. Pour une description complète du système LaTeX, reportez-vous à [1, 3].

LaTeX est disponible pour une vaste gamme de systèmes informatiques, des PCs et Macs aux systèmes UNIX [2] et VMS. Dans de nombreuses universités, il est installé sur le réseau informatique, prêt à être utilisé. L'information nécessaire pour y accéder devrait être fournie dans le *Local Guide* [5]. Si vous avez des difficultés pour commencer, demandez de l'aide à la personne qui vous a donné cette brochure. Ce document *n'est pas* un guide d'installation du système LaTeX. Son but est de vous montrer comment écrire vos documents afin qu'ils puissent être traités par LaTeX.

Cette introduction est composée de cinq chapitres :

Le chapitre 1 présente la structure élémentaire d'un document LaTeX 2_ε. Il vous apprendra également quelques éléments sur l'histoire de LaTeX. Après avoir lu ce chapitre, vous devriez avoir une vue générale de ce qu'est LaTeX, pour pouvoir assimiler ce qui sera présenté dans les chapitres suivants.

Le chapitre 2 entre dans les détails de la mise en page d'un document. Il explique les commandes et les environnements essentiels de LaTeX. Après avoir lu ce chapitre, vous serez capables de rédiger vos premiers documents.

Le chapitre 3 explique comment coder des formules mathématiques avec LaTeX. De nombreux exemples sont donnés pour montrer comment utiliser cet atout majeur de LaTeX. À la fin de ce chapitre, vous trouverez des tableaux qui listent tous les symboles mathématiques disponibles.

Le chapitre 4 explique comment réaliser un index, une liste de références

[2] UNIX est une marque déposée de The Open Group

bibliographiques ou l'insertion de figures PostScript encapsulé. Il présente aussi quelques autres extensions utiles.

Le chapitre 5 contient des informations potentiellement dangereuses. Il vous apprend à modifier la mise en page standard produite par LaTeX et vous permet de transformer les présentations plutôt réussies de LaTeX en quelque chose d'assez laid.

Il est important de lire ces chapitres dans l'ordre. Après tout, ce document n'est pas si long. L'étude attentive des exemples est indispensable à la compréhension de l'ensemble car ils contiennent une bonne partie de l'information que vous pourrez trouver ici.

Si vous avez besoin de récupérer des fichiers relatifs à LaTeX, utilisez l'une des archives ftp `CTAN`. En France elles sont sur le site `ftp://ftp.lip6.fr/pub/TeX/CTAN/`. Aux États-Unis, il s'agit de `ftp://ctan.tug.org/`, en Allemagne de `ftp://ftp.dante.de/` et au Royaume-Uni de `ftp://ftp.tex.ac.uk/`. Si vous n'êtes pas dans l'un de ces pays, choisissez le site le plus proche de chez vous.

Vous verrez plusieurs références à CTAN au long de ce document, en particulier des pointeurs vers des logiciels ou des documents. Au lieu d'écrire des URL complets, nous avons simplement écrit `CTAN:` suivi du chemin d'accès à partir de l'un des sites CTAN ci-dessus.

Si vous souhaitez installer LaTeX sur votre ordinateur, vous trouverez sans doute une version adaptée à votre système sur sur `CTAN:/systems`.

Si vous avez des suggestions concernant ce qui pourrait être ajouté, supprimé ou modifié dans ce document, contactez soit directement l'auteur de la version originale, soit moi-même, le traducteur. Nous sommes particulièrement intéressés par des retours d'utilisateurs débutants en LaTeX au sujet des passages de ce livre qui devraient être mieux expliqués.

Tobias Oetiker `<oetiker@ee.ethz.ch>`
Department of Electrical Engineering,
Swiss Federal Institute of Technology, Zürich.

Matthieu Herrb `<matthieu.herrb@laas.fr>`
Laboratoire d'Analyse et d'Architecture des Systèmes,
Centre National de la Recherche Scientifique, Toulouse.

Table des matières

Merci ! iii

Préface v

1 Ce qu'il faut savoir 1
- 1.1 Le nom de la bête . . . 1
  - 1.1.1 TEX . . . 1
  - 1.1.2 LATEX . . . 1
- 1.2 Les bases . . . 3
  - 1.2.1 Auteur, éditeur et typographe . . . 3
  - 1.2.2 Choix de la mise en page . . . 3
  - 1.2.3 Avantages et inconvénients . . . 4
- 1.3 Fichiers source LATEX . . . 5
  - 1.3.1 Espaces . . . 5
  - 1.3.2 Caractères spéciaux . . . 5
  - 1.3.3 Commandes LATEX . . . 6
  - 1.3.4 Commentaires . . . 6
- 1.4 Structure du fichier source . . . 7
- 1.5 Utilisation typique en ligne de commande . . . 8
- 1.6 La mise en page du document . . . 8
  - 1.6.1 Classes de documents . . . 8
  - 1.6.2 Extensions . . . 11
  - 1.6.3 Styles de page . . . 11
- 1.7 Les fichiers manipulés . . . 13
- 1.8 Gros documents . . . 14

2 Mise en page 15
- 2.1 La structure du document et le langage . . . 15
- 2.2 Sauts de ligne et de page . . . 17
  - 2.2.1 Paragraphes justifiés . . . 17
  - 2.2.2 Césure . . . 18
- 2.3 Chaînes prêtes à l'emploi . . . 19
- 2.4 Caractères spéciaux et symboles . . . 19

2.4.1 Guillemets . . . 19
2.4.2 Tirets . . . 20
2.4.3 Tilde (∼) . . . 20
2.4.4 Symbole degré (°) . . . 20
2.4.5 Points de suspension (...) . . . 21
2.4.6 Ligatures . . . 21
2.4.7 Accents et caractères spéciaux . . . 22
2.5 Support multilingue . . . 22
2.5.1 Support de la langue française . . . 24
2.5.2 Support de la langue allemande . . . 25
2.6 L'espace entre les mots . . . 25
2.7 Titres, chapitres et sections . . . 26
2.8 Références croisées . . . 28
2.9 Notes de bas de page . . . 28
2.10 Souligner l'importance d'un mot . . . 29
2.11 Environnements . . . 30
2.11.1 Listes, énumérations et descriptions . . . 30
2.11.2 Alignements à gauche, à droite et centrage . . . 31
2.11.3 Citations et vers . . . 31
2.11.4 Impression *verbatim* . . . 32
2.11.5 Tableaux . . . 33
2.12 Objets flottants . . . 35
2.13 Protection des commandes « fragiles » . . . 38

3 Formules Mathématiques 39
3.1 Généralités . . . 39
3.2 Groupements en mode mathématique . . . 41
3.3 Éléments d'une formule mathématique . . . 41
3.4 Espacement en mode mathématique . . . 46
3.5 Alignements verticaux . . . 46
3.6 Fantômes... . . . 48
3.7 Taille des polices mathématiques . . . 49
3.8 Insertion de texte en mode mathématique . . . 49
3.9 Théorèmes, propositions, etc. . . . 50
3.10 Symboles gras . . . 51
3.11 Liste des symboles mathématiques . . . 53

4 Compléments 61
4.1 Figures PostScript . . . 61
4.2 Références bibliographiques . . . 63
4.3 Index . . . 64
4.4 En-têtes améliorés . . . 65
4.5 L'extension verbatim . . . 66
4.6 Téléchargement et installation d'extensions . . . 67

5 Personnalisation de LaTeX **69**
5.1 Vos propres commandes, environnements et extensions 69
5.1.1 Nouvelles commandes 70
5.1.2 Nouveaux environnements 71
5.1.3 Votre propre extension 71
5.2 Polices et tailles des caractères 72
5.2.1 Commandes de changement de police 72
5.2.2 Attention danger 75
5.2.3 Un conseil 75
5.3 Espacement 75
5.3.1 Entre les lignes 75
5.3.2 Mise en page d'un paragraphe 76
5.3.3 Espacement horizontal 77
5.3.4 Espacement vertical 77
5.4 Disposition d'une page 78
5.5 Jouons un peu avec les dimensions 80
5.6 Boîtes 81
5.7 Filets 83

Bibliographie **85**

Index **87**

Table des figures

1.1 Éléments d'un système TEX 2
1.2 Un fichier LATEX minimal 8
1.3 Exemple d'un article de revue plus réaliste 9

4.1 Exemple de configuration de l'extension fancyhdr 66

5.1 Exemple d'extension 72
5.2 Paramètres de la disposition d'une page 79

Liste des tableaux

1.1 Classes de documents . . . 9
1.2 Options de classes de document . . . 10
1.3 Quelques extensions fournies avec LaTeX . . . 12
1.4 Les styles de page de LaTeX . . . 13

2.1 Accents et caractères spéciaux . . . 22
2.2 Commandes de saisie en français. . . . 24
2.3 Caractères spéciaux en allemand. . . . 25
2.4 Placements possibles . . . 36

3.1 Accents en mode mathématique . . . 53
3.2 Alphabet grec minuscule . . . 53
3.3 Alphabet grec majuscule . . . 53
3.4 Relations binaires . . . 54
3.5 Opérateurs binaires . . . 54
3.6 Opérateurs n-aires . . . 55
3.7 Flèches . . . 55
3.8 Délimiteurs . . . 55
3.9 Grands délimiteurs . . . 55
3.10 Symboles divers . . . 56
3.11 Symboles non-mathématiques . . . 56
3.12 Délimiteurs de l'AMS . . . 56
3.13 Caractères grecs et hébreux de l'AMS . . . 56
3.14 Relations binaires de l'AMS . . . 57
3.15 Flèches de l'AMS . . . 57
3.16 Négations des relations binaires et des flèches de l'AMS . . . 58
3.17 Opérateurs binaires de l'AMS . . . 58
3.18 Symboles divers de l'AMS . . . 59
3.19 Polices mathématiques . . . 59

4.1 Clefs pour l'extension graphicx . . . 62
4.2 Exemples de clefs d'index . . . 64

5.1 Polices . . . 73
5.2 Tailles des polices . . . 73

5.3 Tailles en points dans les classes standard 73
5.4 Polices mathématiques 74
5.5 Unités TEX 78

Chapitre 1

Ce qu'il faut savoir

Dans la première partie de ce chapitre vous trouverez une rapide présentation de la philosophie et de l'histoire de LaTeX 2$_\varepsilon$. La deuxième partie met l'accent sur les structures fondamentales d'un document LaTeX. Après avoir lu ce chapitre, vous devriez avoir une idée d'ensemble du fonctionnement de LaTeX qui vous aidera à mieux comprendre les chapitres suivants.

1.1 Le nom de la bête

1.1.1 TeX

TeX est un programme écrit par Donald E. Knuth [2]. Il est conçu pour la composition de textes et d'équations mathématiques.

Knuth a commencé le développement de TeX en 1977 parce qu'il était frustré par la manière avec laquelle ses articles étaient publiés par l'American Mathematical Society. Il avait arrêté de soumettre des articles vers 1974 parce que « le résultat final était trop pénible à regarder ». TeX, tel que nous l'utilisons aujourd'hui, est sorti en 1982 et a été amélioré au fil des ans. Ces dernières années TeX a atteint une grande stabilité. Aujourd'hui Knuth affirme qu'il n'y a virtuellement plus de « bug ». Le numéro de version de TeX tend vers π et est actuellement 3.14159.

TeX se prononce « Tech », avec un « ch » comme dans le mot écossais « Loch ». En alphabet phonétique cela donne **[tex]**. . .Dans un environnement `ASCII`, TeX devient `TeX`.

1.1.2 LaTeX

LaTeX est un ensemble de macros qui permettent à un auteur de mettre en page son travail avec la meilleure qualité typographique en utilisant un format professionnel pré-défini. LaTeX a été écrit par Leslie Lamport [1]. Il utilise TeX comme outil de mise en page.

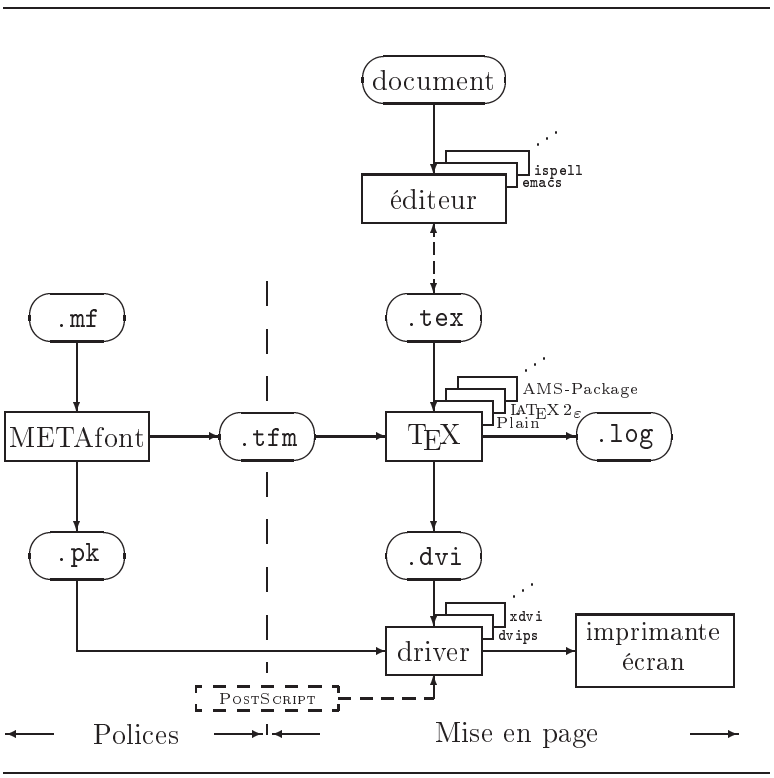


FIG. 1.1 – Éléments d'un système TEX

En 1994, LATEX a été mis à jour par l'équipe LATEX3, menée par Frank Mittelbach, afin de réaliser certaines améliorations demandées depuis longtemps et de fusionner toutes les variantes qui s'étaient développées depuis la sortie de LATEX 2.09 quelques années auparavant. Pour distinguer cette nouvelle version des précédentes, elle est appelée LATEX 2_ε. Ce document est relatif à LATEX 2_ε.

LATEX se prononce **[latex]**. Si vous voulez faire référence à LATEX dans un environnement `ASCII`, utilisez `LaTeX`. LATEX 2_ε se prononce **[latex døzø]** et s'écrit `LaTeX2e`.

En anglais, cela donne **[laɪtex]** et **[laɪtex tu: i:]**.

La figure 1.1, page 2 montre l'interaction entre les différents éléments d'un système TEX. Cette figure est extraite de `wots.tex` de Kees van der Laan.

1.2 Les bases

1.2.1 Auteur, éditeur et typographe

Pour publier un texte, un auteur confie son manuscrit à une maison d'édition. L'éditeur décide alors de la mise en page du document (largeur des colonnes, polices de caractères, présentation des en-têtes,...). L'éditeur note ses instructions sur le manuscrit et le passe à un technicien typographe qui réalise la mise en page en suivant ces instructions.

Un éditeur humain essaye de comprendre ce que l'auteur veut mettre en valeur et décide de la présentation en fonction de son expérience professionnelle et du contenu du manuscrit.

Dans un environnement LaTeX, celui-ci joue le rôle de l'éditeur et utilise TeX comme typographe pour la composition. Mais LaTeX n'est qu'un programme et a donc besoin de plus de directives. L'auteur doit en particulier lui fournir la structure logique de son document. Cette information est insérée dans le texte sous la forme de « commandes LaTeX ».

Cette approche est totalement différente de l'approche WYSIWYG[1] utilisée par les traitements de texte modernes tels que *Microsoft Word* ou *Corel WordPerfect*. Avec ces programmes, l'auteur définit la mise en page du document de manière interactive pendant la saisie du texte. Tout au long de cette opération, il voit à l'écran à quoi ressemblera le document final une fois imprimé.

Avec LaTeX, il n'est normalement pas possible de voir le résultat final durant la saisie du texte. Mais celui-ci peut être pré-visualisé après traitement du fichier par LaTeX. Des corrections peuvent alors être apportées avant d'envoyer la version définitive vers l'imprimante.

1.2.2 Choix de la mise en page

La typographie est un métier (un art ?). Les auteurs inexpérimentés font souvent de graves erreurs en considérant que la mise en page est avant tout une question d'esthétique : « si un document est beau, il est bien conçu ». Mais un document doit être lu et non accroché dans une galerie d'art. La lisibilité et la compréhensibilité sont bien plus importantes que le « look ». Par exemple :

- la taille de la police et la numérotation des en-têtes doivent être choisies afin de mettre en évidence la structure des chapitres et des sections ;
- les lignes ne doivent pas être trop longues pour ne pas fatiguer la vue du lecteur, tout en remplissant la page de manière harmonieuse.

Avec un logiciel WYSIWYG, l'auteur produit généralement des documents esthétiquement plaisants (quoi que...) mais très peu ou mal structurés. LaTeX empêche de telles erreurs de formatage en forçant l'auteur à

[1] What you see is what you get – Ce que vous voyez est ce qui sera imprimé.

décrire la structure logique de son document et en choisissant lui-même la mise en page la plus appropriée.

1.2.3 Avantages et inconvénients

Un sujet de discussion qui revient souvent quand des gens du monde WYSIWYG rencontrent des utilisateurs de LaTeX est le suivant : « les avantages de LaTeX par rapport à un traitement de texte classique » ou bien le contraire. La meilleure chose à faire quand une telle discussion démarre, est de garder son calme, car souvent cela dégénère. Mais parfois on ne peut y échapper...

Voici donc quelques arguments. Les principaux avantages de LaTeX par rapport à un traitement de texte traditionnel sont :

- mise en page professionnelle qui donne aux documents l'air de sortir de l'atelier d'un imprimeur ;
- la composition des formules mathématiques se fait de manière pratique ;
- il suffit de connaître quelques commandes de base pour décrire la structure logique du document. Il n'est pas nécessaire de se préoccuper de la mise en page ;
- des structures complexes telles que des notes de bas de page, des renvois, la table des matières ou les références bibliographiques sont produites facilement ;
- pour la plupart des tâches de la typographie qui ne sont pas directement gérées par LaTeX, il existe des extensions gratuites. Par exemple pour inclure des figures PostScript ou pour formater une bibliographie selon un standard précis. La majorité de ces extensions sont décrites dans *The LaTeX Companion* [3] (en anglais) et dans *LaTeX, Apprentissage, guide et référence* [4] (en français) ;
- LaTeX encourage les auteurs à écrire des documents bien structurés, parce que c'est ainsi qu'il fonctionne (en décrivant la structure) ;
- TeX, l'outil de formatage de LaTeX 2_ε, est réellement portable et gratuit. Ainsi il est disponible sur quasiment toutes les machines existantes.

LaTeX a également quelques inconvénients ; il est difficile pour moi d'en trouver, mais d'autres vous en citeront des centaines :

- LaTeX ne fonctionne pas bien pour ceux qui ont vendu leur âme ;
- bien que quelques paramètres des mises en page pré-définies puissent être personnalisés, la mise au point d'une présentation entièrement nouvelle est difficile et demande beaucoup de temps [2] ;
- écrire des documents mal organisés et mal structurés est très difficile.

[2] La rumeur dit que c'est un des points qui devrait être améliorés dans la future version LaTeX3

1.3 Fichiers source LaTeX

L'entrée de LaTeX est un fichier texte `ASCII`. Vous pouvez le créer avec l'éditeur de texte de votre choix. Il contient le texte de votre document ainsi que les commandes qui vont dire à LaTeX comment mettre en page le texte. On appelle ce fichier *fichier source*.

1.3.1 Espaces

Les caractères d'espacement, tels que les blancs ou les tabulations sont traités de manière unique comme « espace » par LaTeX. Plusieurs blancs *consécutifs* sont considérés comme *une seule* espace [3]. Les espaces en début d'une ligne sont en général ignorées et un seul retour à la ligne est traité comme une espace.

Une ligne vide entre deux lignes de texte marque la fin d'un paragraphe. *Plusieurs* lignes vides sont considérées comme *une seule* ligne vide. Le texte ci-dessous est un exemple. Sur la gauche se trouve le contenu du fichier source et à droite le résultat formaté.

```
Saisir un ou      plusieurs
espaces entre  les     mots
n'a pas d'importance.

Une ligne vide commence
un nouveau paragraphe.
```

Saisir un ou plusieurs espaces entre les mots n'a pas d'importance.
Une ligne vide commence un nouveau paragraphe.

1.3.2 Caractères spéciaux

Les symboles suivants sont des caractères réservés qui, soit ont une signification spéciale dans LaTeX, soit ne sont pas disponibles dans toutes les polices. Si vous les saisissez directement dans votre texte, ils ne seront pas imprimés et forceront LaTeX à faire des choses que vous n'avez pas voulues.

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

Comme vous le voyez, certains de ces caractères peuvent être utilisés dans vos documents en les préfixant par un antislash :

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

$ & % # _ { }

Les autres et bien d'autres encore peuvent être obtenus avec des commandes spéciales à l'intérieur de formules mathématiques ou comme accents.

[3] En langage typographique, *espace* est un mot féminin. *NdT.*

L'antislash \ ne peut pas être saisi en ajoutant un second antislash (\\). Cette séquense est utilisée pour indiquer les coupures de ligne[4].

1.3.3 Commandes LaTeX

Les commandes LaTeX sont sensibles à la casse des caractères (majuscules ou minuscules) et utilisent l'un des deux formats suivants :

- soit elles commencent par un antislash \ et ont un nom composé uniquement de lettres. Un nom de commande est terminé par une espace, un chiffre ou tout autre caractère qui n'est pas une lettre ;
- soit elles sont composées d'un antislash et d'un caractère spécial (non-lettre) exactement.

LaTeX ignore les espaces après les commandes. Si vous souhaitez obtenir un blanc après une commande, il faut ou bien insérer `{}` suivi d'un blanc ou bien utiliser une commande d'espacement spécifique de LaTeX. La séquence `{}` empêche LaTeX d'ignorer les blancs après une commande.

```
J'ai lu que Knuth classe les
gens qui utilisent \TeX{} en
\TeX{}niciens et en \TeX perts.\\
Aujourd'hui nous sommes le \today.
```

J'ai lu que Knuth classe les gens qui utilisent TeX en TeXniciens et en TeXperts.
Aujourd'hui nous sommes le 25 novembre 2001.

Certaines commandes sont suivies d'un paramètre qui est mis entre accolades `{ }`. Certaines commandes supportent des paramètres optionnels qui suivent le nom de la commande entre crochets `[ ]`. L'exemple suivant montre quelques commandes LaTeX. Ne vous tracassez pas pour les comprendre, elles seront expliquées plus loin.

```
\textsl{Penchez}-vous !
```

Penchez-vous !

```
S'il vous plait, passez \'a la
ligne ici.\newline
Merci !
```

S'il vous plait, passez à la ligne ici.
Merci !

1.3.4 Commentaires

Quand LaTeX rencontre un caractère `%` dans le fichier source, il ignore le reste de la ligne en cours. C'est utile pour ajouter des notes qui n'apparaîtront pas dans la version imprimée.

[4] Utilisez la commande `$\backslash$`. Elle produit un \.

```
% Demonstration :
Ceci est un % mauvais
exemple: anticonstitu%
       tionnellement
```

Ceci est un exemple: anticonstitutionnellement

Le caractère % peut également être utilisé pour couper des lignes trop longues dans le fichier d'entrée, lorsqu'aucun espace ou coupure n'est autorisé.

Pour créer des commentaires plus longs, il vaut mieux utilier l'environnement `comment` fourni par l'extension `verbatim`. Vous apprendrez plus loin à utiliser une extension.

```
Voici un autre exemple
\begin{comment}
Limité mais demonstratif
\end{comment}
de commentaires.
```

Voici un autre exemple de commentaires.

Cet environnement n'est pas utilisable à l'intérieur d'autres environnements complexes, tels que le mode mathématique par exemple.

1.4 Structure du fichier source

Quand LaTeX analyse un fichier source, il s'attend à y trouver une certaine structure. C'est pourquoi chaque fichier source doit commencer par la commande :

```
\documentclass{...}
```

Elle indique quel type de document vous voulez écrire. Après cela vous pouvez insérer des commandes qui vont influencer le style du document ou vous pouvez charger des extensions qui ajoutent de nouvelles fonctions au système LaTeX. Pour charger une extension, utilisez la commande :

```
\usepackage{...}
```

Quand tout le travail de préparation est fait[5], vous pouvez commencer le corps du texte avec la commande :

```
\begin{document}
```

Maintenant vous pouvez saisir votre texte et y insérer des commandes LaTeX. À la fin de votre document, utilisez la commande

```
\end{document}
```

pour dire à LaTeX qu'il en a fini. Tout ce qui suivra dans le fichier source sera ignoré.

La figure 1.2 montre le contenu d'un document LaTeX 2_ε minimum. Un fichier source plus complet est présenté sur la figure 1.3.

[5] La partie entre `\documentclass` et `\begin{document}` est appelée le *préambule*.

1.5 Utilisation typique en ligne de commande

Vous brûlez probablement d'envie d'essayer l'exemple présenté page 8. Voici quelques informations : LaTeX lui-même ne propose pas d'interface graphique ni de jolis boutons à cliquer. Il s'agit simplement d'un programme qui « digère » votre fichier source. Certaines installations de LaTeX ajoutent une telle interface graphique permettant de cliquer pour lancer la mise en page de votre document.

Mais les puristes ne cliquent pas ; voici donc comment demander à LaTeX de traiter votre document sur un système avec une interface textuelle. Remarquez que ceci suppose que LaTeX soit déjà installé sur votre ordinateur.

1. Créez/éditez votre fichier source LaTeX. Il s'agit d'un fichier texte pur. Sur les systèmes Unix, tous les éditeurs créent ce type de fichier. Sous Windows, assurez-vous que le fichier est sauvegardé en texte seul (ASCII ou encore « plain text »). Choisissez pour votre fichier un nom avec le suffixe `.tex`.
2. Exécutez LaTeX sur votre fichier. Si tout se passe bien, vous obtiendrez un nouveau fichier avec le suffixe `.dvi`.

   ```
   latex document.tex
   ```
3. À présent, vous pouvez visualiser le résultat, le fichier DVI.

   ```
   xdvi document.dvi
   ```

   ou encore transformer le résultat en PostScript

   ```
   dvips -Pcmz -o document.ps document.dvi
   ```

   `xdvi` et `dvips` sont des logiciels qui manipulent les fichiers DVI. Ils sont disponibles sur la plupart des systèmes Unix. Sur les autres systèmes, d'autres outils de manipulation des fichiers DVI sont disponibles.

1.6 La mise en page du document

1.6.1 Classes de documents

La première information dont LaTeX a besoin en traitant un fichier source est le type de document que son auteur est en train de créer. Ce type est

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

Fig. 1.2 – Un fichier LaTeX minimal

```
\documentclass[a4paper,11pt]{article}
\usepackage[T1]{fontenc}
\usepackage[english,francais]{babel}
\author{P.~Tar}
\title{Le Minimalisme}
\begin{document}
\maketitle
\tableofcontents
\section{D\'ebut}
\`A \'ecrire\dots
\section{Suite et fin}
On verra plus tard.
\end{document}
```

FIG. 1.3 – Exemple d'un article de revue plus réaliste

spécifié par la commande `\documentclass`.

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

Ici *classe* indique le type de document à créer. Le tableau 1.1 donne la liste des classes de documents présentées dans cette introduction. LaTeX 2ε fournit d'autres classes pour d'autres types de documents, notamment des lettres et des transparents. Le paramètre *options* permet de modifier le comportement de la classe de document. Les options sont séparées par des virgules. Les principales options disponibles sont présentées dans le tableau 1.2.

TAB. 1.1 – Classes de documents

`article`	pour des articles dans des revues scientifiques, des présentations, des rapports courts, des documentations, des invitations, etc.
`report`	pour des rapports plus longs contenant plusieurs chapitres, des petits livres, des thèses, etc.
`book`	pour des vrais livres.
`slides`	pour des transparents. Cette classe utilise de grands caractères sans serif. Voir également la classe FoilTeX [a]

[a] `CTAN:/tex-archive/macros/latex/contrib/supported/foiltex`

Exemple : un fichier source pour un document LaTeX pourrait commencer par la ligne

TAB. 1.2 – Options de classes de document

`10pt`, `11pt`, `12pt` définit la taille de la police principale du document. Si aucune option n'est présente, la taille par défaut est de `10pt`.

`a4paper`, `letterpaper`, ... définit la taille du papier. Le papier par défaut est `letterpaper`, le format standard américain. Les autres valeurs possibles sont : `a5paper`, `b5paper`, `executivepaper`, et `legalpaper`.

`fleqn` aligne les formules mathématiques à gauche au lieu de les centrer.

`leqno` place la numérotation des formules à gauche plutôt qu'à droite.

`titlepage`, `notitlepage` indique si une nouvelle page doit être commencée après le titre du document ou non. La classe `article` continue par défaut sur la même page contrairement aux classes `report` et `book`.

`twocolumn` demande à LaTeX de formater le texte sur deux colonnes.

`twoside, oneside` indique si la sortie se fera en recto-verso ou en recto simple. Par défaut, les classes `article` et `report` sont en simple face alors que la classe `book` est en double-face.

`openright, openany` fait commencer un chapitre sur la page de droite ou sur la prochaine page. Cette option n'a pas de sens avec la classe `article` qui ne connaît pas la notion de chapitre. Par défaut, la classe `report` commence les chapitres sur la prochaine page vierge alors que la classe `book` les commence toujours sur une page de droite.

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

elle informe LaTeX qu'il doit composer ce document comme un *article* avec une taille de caractère de base de *onze points* et qu'il devra produire une mise en page pour une impression *double face* sur du papier au format *A4* [6].

1.6.2 Extensions

En rédigeant votre document, vous remarquerez peut-être qu'il y a des domaines où les commandes de base de LaTeX ne permettent pas d'exprimer ce que vous voudriez. Si vous voulez inclure des graphiques, du texte en couleur ou du code d'un programme dans votre document, il faut augmenter les possibilités de LaTeX grâce à des extensions. Une extension est chargée par la commande

`\usepackage[`*options*`]{`*extension*`}`

extension est le nom de l'extension et *options* une liste de mots-clés qui déclenchent certaines fonctions de l'extension. Certaines extensions font partie de la distribution standard de LaTeX 2$_\varepsilon$ (reportez-vous au tableau 1.3). D'autres sont distribuées à part. Le *Local Guide* [5] peut vous fournir plus d'informations sur les extensions installées sur votre site. *The LaTeX Companion* [3] est la principale source d'information au sujet de LaTeX 2$_\varepsilon$. Ce livre contient la description de centaines d'extensions ainsi que les informations nécessaires pour écrire vos propres extensions à LaTeX 2$_\varepsilon$.

1.6.3 Styles de page

LaTeX propose trois combinaisons d'en-têtes et de pieds de page, appelées styles de page et définies par le paramètre *style* de la commande :

`\pagestyle{`*style*`}`

Le tableau 1.4 donne la liste des styles prédéfinis.

On peut changer le style de la page en cours grâce à la commande

`\thispagestyle{`*style*`}`

Au chapitre 4, page 65, vous apprendrez comment créer vos propres entêtes et pieds de pages.

[6] Sans l'option `a4paper`, le format de papier sera américain : 8,5 × 11 pouces, soit 216 × 280 mm.

TAB. 1.3 – Quelques extensions fournies avec LaTeX

doc permet de documenter des programmes pour LaTeX.
Décrite dans `doc.dtx`[a] et dans *The LaTeX Companion* [3].

exscale fournit des versions de taille paramétrable des polices mathématiques étendues.
Décrite dans `ltexscale.dtx`.

fontenc spécifie le codage des polices de caractère que LaTeX va utiliser.
Décrite dans `ltoutenc.dtx`.

ifthen fournit des commandes de la forme
'if...then do...otherwise do...'
Décrite dans `ifthen.dtx`, dans *The LaTeX Companion* [3] et dans *LaTeX, Apprentissage, guide et référence* [4].

latexsym permet l'utilisation de la police des symboles LaTeX.
Décrite dans `latexsym.dtx`, dans *The LaTeX Companion* [3] et dans *LaTeX, Apprentissage, guide et référence* [4].

makeidx fournit des commandes pour réaliser un index.
Décrite dans ce document, section 4.3, dans *The LaTeX Companion* [3] et dans *LaTeX, Apprentissage, guide et référence* [4].

syntonly analyse un document sans le formater.
Décrite dans `syntonly.dtx` et dans *The LaTeX Companion* [3]. Utile pour une vérification rapide de la syntaxe.

inputenc permet de spécifier le codage des caractères utilisé dans le fichier source, parmi ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows ou un codage défini par l'utilisateur.
Décrite dans `inputenc.dtx`.

[a] Ce fichier devrait être intallé sur votre système et vous devriez être capable de le formater avec la commande `latex doc.dtx`. Il en est de même pour les autres fichiers cités dans ce tableau.

1.7 Les fichiers manipulés

L'utilisateur de LaTeX est amené à cotoyer un grand nombre de fichiers aux suffixes divers, chaque suffixe renseigne sur le rôle du fichier, il est donc utile d'en connaître la signification, voici les suffixes les plus courants :

.tex fichier source TeX ou LaTeX ;

.sty fichier contenant des commandes, que l'on charge dans le préambule d'un document grâce à une commande `\usepackage` ;

.dtx fichier contenant du code LaTeX (commandes) documenté, le lancement de LaTeX sur un fichier `.dtx` en extrait la documentation.

.ins fichier permettant d'installer le contenu du fichier `.dtx` de même nom. Une extension LaTeX téléchargée de l'Internet est composée d'un fichier `.dtx` et d'un `.ins`. Exécuter LaTeX sur le fichier `.ins` pour extraire les fichiers à installer du `.dtx`.

.cls désigne un fichier de *classe* contenant la description d'un type de document, chargé par la commande `\documentclass`;

.fd fichier contenant des définitions pour les polices de caractères ;

Les fichiers suivants sont produits par LaTeX à partir du fichier source (de suffixe `.tex`) :

.dvi signifie *DeVice Independent*, c'est le fichier que l'on visualise à l'écran et qui servira à l'impression (par `dvips` par exemple) ;

.log fichier contenant le compte-rendu détaillé de la compilation du fichier source (avec les messages d'erreur éventuels),

.toc contient le matériel nécessaire à la production de la table des matières, si celle-ci a été demandée ;

.lof contient la liste numérotée des figures, si elle a été demandée ;

.lot contient la liste numérotée des tableaux, si elle a été demandée ;

.aux contient diverses informations utiles à LaTeX, en particulier ce qui est nécessaire au fonctionnement des références croisées. Le fichier `.aux` produit lors d'une exécution de LaTeX est lu lors de l'exécution suivante ;

Tab. 1.4 – Les styles de page de LaTeX

`plain`	imprime le numéro de page au milieu du pied de page. C'est le style par défaut.
`headings`	imprime le titre du chapitre courant et le numéro de page dans l'en-tête de chaque page et laisse le pied de page vide. C'est à peu près le style utilisé dans ce document.
`empty`	laisse l'en-tête et le pied de page vides.

.idx fichier produit seulement si un index est demandé, il doit être traité par `makeindex` (voir section 4.3 page 64) ;

.ind fichier produit par `makeindex` à partir du `.idx`, il contient l'index prêt à être inclus dans le document ;

.ilg fichier contenant le compte-rendu du travail de `makeindex`.

1.8 Gros documents

Lorsque l'on travaille sur de gros documents, il peut être pratique de couper le fichier source en plusieurs morceaux. LaTeX a deux commandes qui vous permettent de gérer plusieurs fichiers sources.

`\include{`*fichier*`}`

Vous pouvez utiliser cette commande dans le corps de votre document pour insérer le contenu d'un autre fichier source. LaTeX ajoute automatiquement le suffixe `.tex` au nom spécifié. Remarquez que LaTeX va sauter une page pour traiter le contenu de *fichier*`.tex`.

La seconde commande peut être utilisée dans le préambule. Elle permet de dire à LaTeX de n'inclure que certains des fichiers désignés par les commandes `\include`.

`\includeonly{`*fichier*`,`*fichier*`,...}`

Après avoir rencontré cette commande dans le préambule d'un document, seules les commandes `\include` dont les fichiers sont cités en paramètre de la commande `\includeonly` seront exécutées. Attention, il ne doit pas y avoir d'espace entre les noms de fichiers et les virgules.

La commande `\include` saute une page avant de commencer le formatage du texte inclus. Ceci est utile lorsqu'on utilise `\includeonly`, parce qu'ainsi les sauts de pages ne bougeront pas, même si certains morceaux ne sont pas inclus. Parfois ce comportement n'est pas souhaitable. Dans ce cas, vous pouvez utiliser la commande :

`\input{`*fichier.tex*`}`

qui insère simplement le fichier indiqué sans aucun traitement sophistiqué.

Il est possible de demander à LaTeX de simplement vérifier la syntaxe d'un document, sans produire de fichier `.dvi` pour gagner du temps, en utilisant l'extension `syntonly` :

```
\usepackage{syntonly}
\syntaxonly
```

La vérification terminée, il suffit de mettre ces deux lignes (ou simplement la seconde) en commentaire en plaçant un `%` en tête de ligne.

Chapitre 2

Mise en page

Après la lecture du chapitre précédent vous connaissez maintenant les éléments de base qui constituent un document LaTeX. Dans ce chapitre, nous allons compléter vos connaissances afin de vous rendre capables de créer des documents réalistes.

2.1 La structure du document et le langage

La principale raison d'être d'un texte (à l'exception de certains textes de la littérature contemporaine) est de diffuser des idées, de l'information ou de la connaissance au lecteur. Celui-ci comprendra mieux le texte si ces idées sont bien structurées et il ressentira d'autant mieux cette structure si la typographie utilisée reflète la structure logique et sémantique du contenu.

Ce qui distingue LaTeX des autres logiciels de traitement de texte c'est qu'il suffit d'indiquer à LaTeX la structure logique et sémantique d'un texte. Il en déduit la forme typographique en fonction des « règles » définies dans la classe de document et les différents fichiers de style.

L'élément de texte le plus important pour LaTeX (et en typographie) est le paragraphe. Le paragraphe est la forme typographique qui contient une pensée cohérente ou qui développe une idée. Vous allez apprendre dans les pages suivantes la différence entre un retour à la ligne (obtenu avec la commande \\) et un changement de paragraphe (obtenu en laissant une ligne vide dans le document source). Une nouvelle réflexion doit débuter sur un nouveau paragraphe, mais si vous poursuivez une réflexion déjà entamée, un simple retour à la ligne suffit.

En général, on sous-estime complètement l'importance du découpage en paragraphes. Certains ignorent même la signification d'un changement de paragraphe ou bien, notamment avec LaTeX, coupent des paragraphes sans le savoir. Cette erreur est particulièrement fréquente lorsque des équations sont présentes au milieu du texte. Étudiez les exemples suivants et essayez de comprendre pourquoi des lignes vides (changements de paragraphe) sont

parfois utilisées avant et après l'équation et parfois non. (Si vous ne comprenez pas suffisamment les commandes utilisées, lisez d'abord la suite du chapitre puis revenez à cette section.)

```
% Exemple 1
\dots{} lorsqu'Einstein introduit sa formule
\begin{equation}
  e = m \cdot c^2 \; ,
\end{equation}
qui est en même temps la formule la plus connue et la
moins comprise de la physique.

% Exemple 2
\dots{} d'où vient la loi des courants de Kirchhoff :
\begin{equation}
  \sum_{k=1}{n} I_k = 0 \; .
\end{equation}

La loi des tensions de Kirchhof s'en déduit\dots

% Exemple 3
\dots{} qui a plusieurs avantages.

\begin{equation}
  I_D = I_F - I_R
\end{equation}
est le c\oe{}ur d'un modèle de transistor très
différent\dots
```

L'unité de texte immédiatement inférieure est la phrase. Dans les documents anglo-saxons, l'espace après le point terminant une phrase est plus grande que celle qui suit un point après une abréviation. (Ceci n'est pas vrai dans les règles de la typographie française). En général, LaTeX se débrouille pour déterminer la bonne largeur des espaces. S'il n'y arrive pas, vous verrez dans la suite comment le forcer à faire quelquechose de correct.

La structure du texte s'étend même aux morceaux d'une phrase. Les règles grammaticales de chaque langue gèrent la ponctuation de manière très précise. Dans la plupart des langues, la virgule représente une courte respiration dans le flux du langage. Si vous ne savez pas trop où placer une virgule, lisez la phrase à voix haute en respirant à chaque virgule. Si cela ne sonne pas naturellement à certains endroits, supprimez la virgule. Au contraire, si vous ressentez le besoin de respirer (ou de marquer une courte pause), insérez un virgule à cet endroit.

Enfin, les paragraphes d'un texte sont également structurés au niveau supérieur, en les regroupant en sections, chapitres, etc. L'effet typographique

d'une commande telle que

```
\section{La structure du texte et du langage}
```

est suffisament évident pour comprendre comment utiliser ces structures de haut niveau.

2.2 Sauts de ligne et de page

2.2.1 Paragraphes justifiés

Bien souvent les livres sont composés de lignes qui ont toutes la même longueur ; on dit qu'elles sont justifiées à droite. LaTeX insère des retours à la ligne et des espacements entre les mots de manière à optimiser la présentation de l'ensemble d'un paragraphe. En cas de besoin, il coupe les mots qui ne tiennent pas en entier sur une ligne. La présentation exacte d'un paragraphe dépend de la classe de document [1]. Normalement la première ligne d'un paragraphe est en retrait par rapport à la marge gauche et il n'y a pas d'espace vertical particulière entre deux paragraphes (cf. section 5.3.2).

Dans certains cas particuliers, il peut être nécessaire de demander à LaTeX de couper une ligne :

`\\` ou `\newline`

commence une nouvelle ligne sans commencer un nouveau paragraphe.

`\\*`

empêche un saut de page après le saut de ligne demandé.

`\newpage`

provoque un saut de page.

`\linebreak[`*n*`]`, `\nolinebreak[`*n*`]`, `\pagebreak[`*n*`]` et `\nopagebreak[`*n*`]`

font ce que suggère leur nom anglais. Ces commandes permettent à l'auteur de paramétrer leur action par l'intermédiaire du paramètre optionnel *n*. Il peut prendre une valeur entre zéro et quatre. En donnant à *n* une valeur inférieure à quatre, vous laissez à LaTeX la possibilité de ne pas tenir compte de votre commande si cela devait rendre le résultat réellement laid. Ne confondez pas ces commandes « break » avec les commandes « new ». Même lorsque vous utilisez une commande « break », LaTeX essaye de justifier le bord droit du texte et d'ajuster la longueur totale de la page, comme expliqué plus

[1] et des règles typographiques propres de chaque pays *NdT*.

loin. Si vous voulez réellement commencer une « nouvelle » ligne, utilisez la commande « new » correspondante.

LaTeX essaye toujours de trouver les meilleurs endroits pour les retours à la ligne. S'il ne trouve pas de solution pour couper les lignes de manière conforme à ses normes de qualité, il laisse dépasser un bout de ligne sur la marge droite du paragraphe. LaTeX émet alors le message d'erreur « overfull hbox[2] ». Cela se produit surtout quand LaTeX ne trouve pas de point de césure dans un mot. En utilisant alors la commande `\sloppy`, vous pouvez demander à LaTeX d'être moins exigeant. Il ne produira plus de lignes trop longues en ajoutant de l'espace entre les mots du paragraphe, même si ceux-ci finissent trop espacés selon ses critères. Dans ce cas le message « underfull hbox[3] » est produit. Souvent, malgré tout, le résultat est acceptable. La commande `\fussy` agit dans l'autre sens, au cas où vous voudriez voir LaTeX revenir à ses exigences normales.

2.2.2 Césure

LaTeX coupe les mots en fin de ligne si nécessaire. Si l'algorithme de césure[4] ne trouve pas l'endroit correct pour couper un mot[5], vous pouvez utiliser les commandes suivantes pour informer TeX de l'exception.

La commande :

`\hyphenation{`*liste de mots*`}`

permet de ne couper les mots cités en argument qu'aux endroits indiqués par « - ». Cette commande doit être placée dans le préambule et ne doit contenir que des mots composés de lettres normales. La casse des caractères n'est pas prise en compte. L'exemple ci-dessous permet à « anticonstitutionnellement » d'être coupé, ainsi qu'à « Anticonstitutionnellement ». Mais il empêche toute césure de « FORTRAN », « Fortran » ou « fortran ». Ni les caractères spéciaux ni les symboles ne sont autorisés dans cette commande.

```
\hyphenation{FORTRAN}
\hyphenation{Anti-cons-ti-tu-tion-nel-le-ment}
```

La commande `\hyphenation{`*liste de mots*`}` a un effet *global* sur toutes les occurrences des mots de la liste. Si l'exception ne concerne qu'une occurrence d'un mot on utilise la commande `\-` qui insère un point de césure potentiel dans un mot. Ces positions deviennent alors les *seuls* points de césure possibles pour cette occurrence du mot.

[2] débordement horizontal

[3] boîte horizontale pas assez pleine

[4] *Hyphenation* en anglais

[5] Ce qui est normalement plutôt rare. Si vous observez de nombreuses erreurs de césure, c'est probablement un problème de spécification de la langue du document ou du codage de sortie. Voir le paragraphe sur le support multilingue, page 22.

```
I think this is: su\-per\-cal\-%
i\-frag\-i\-lis\-tic\-ex\-pi\-%
al\-i\-do\-cious
```

I think this is: supercalifragilisticexpialidocious

Normalement, en français, on ne coupe pas la dernière syllabe d'un mot si elle est muette, mais il arrive qu'on soit obligé de le faire, par exemple si on travaille sur des textes étroits (cas de colonnes multiples).

Exemple: on pourra coder `ils ex\-pri\-ment` pour autoriser *exception-nellement* le rejet à la ligne suivante de la syllabe muette `ment`.

Plusieurs mots peuvent être maintenus ensemble sur une ligne avec la commande :

`\mbox{`*texte*`}`

Elle a pour effet d'interdire toute coupure de ligne dans *texte*.

```
Mon num\'ero de t\'el\'ephone va
changer. \`A partir du 18 octobre,
ce sera le  \mbox{0561 336 330}.

Le param\`etre
\mbox{\emph{nom du fichier}}
de la commande \texttt{input}
contient le nom du fichier
\`a lire.
```

Mon numéro de téléphone va changer. À partir du 18 octobre, ce sera le 0561 336 330. Le paramètre *nom du fichier* de la commande `input` contient le nom du fichier à lire.

2.3 Chaînes prêtes à l'emploi

Dans les exemples précédents, vous avez découvert certaines commandes permettant de produire le logo LaTeX et quelques autres chaînes de caractères spécifiques. Voici une liste de quelques-unes de ces commandes :

Commande	Résultat	Description
`\today`	25 novembre 2001	Date du jour dans la langue courante
`\TeX`	TeX	Logo TeX
`\LaTeX`	LaTeX	Logo LaTeX
`\LaTeXe`	LaTeX 2$_\varepsilon$	Version actuelle de LaTeX

2.4 Caractères spéciaux et symboles

2.4.1 Guillemets

Pour insérer des guillemets n'utilisez pas le caractère " comme sur une machine à écrire. En typographie, il y a des guillemets ouvrants et fermants

spécifiques. En anglais, utilisez deux ‘ pour les guillemets ouvrants et deux ’ pour les guillemets fermants. En français, avec l'option francais de l'extension babel, utilisez \og et \fg ou bien utilisez directement « et » si vous disposez d'un moyen de saisir ces caractères.

```
‘‘Please press the ‘x’ key.’’

«~Appuyez sur la touche ‘x’.~»
```

“Please press the ‘x’ key.”
« Appuyez sur la touche ‘x’. »

2.4.2 Tirets

LaTeX connaît quatre types de tirets. Trois d'entre eux sont obtenus en juxtaposant un nombre variable de tirets simples. Le quatrième n'est pas réellement un tiret, il s'agit du signe mathématique moins.

```
belle-fille, pages 13-67\\
il parle ---~en vain~---
du passé.\\
oui~---~ou non ? \\
$0$, $1$ et $-1$
```

belle-fille, pages 13-67
il parle — en vain — du passé.
oui — ou non ?
0, 1 et −1

Notez que les exemples ci-dessus respectent les règles de la typographie française concernant l'usage des tirets, qui diffèrent des habitudes anglo-saxonnes, en particulier le double tiret n'est pas utilisé en français.

2.4.3 Tilde (∼)

Un caractère souvent utilisé dans les adresses sur le web est le tilde. Pour produire ce caractère avec LaTeX, on peut utiliser \~, mais le résultat : ~ n'est pas tout à fait le symbole attendu. Essayez ceci à la place :

```
http://www.rich.edu/\~{}bush \\
http://www.clever.edu/$\sim$demo
```

http://www.rich.edu/~bush
http://www.clever.edu/∼demo

Voir aussi l'extension hyperref qui inclut une commande \url.

2.4.4 Symbole degré (°)

Comment obtenir un symbole degré avec LaTeX ?

```
Il fait $-30\,^{\circ}\mathrm{C}$,
je vais bient\^ot devenir
supra-conducteur.
```

Il fait −30 °C, je vais bientôt devenir supra-conducteur.

L'extension babel avec l'option francais introduit la commande `\degres` qui donne un meilleur résultat :

```
Il fait -30~\degres C,
je vais bient\^ot devenir
supra-conducteur.
```

Il fait -30 °C, je vais bientôt devenir supra-conducteur.

2.4.5 Points de suspension (...)

Sur une machine à écrire, une virgule ou un point occupent la même largeur que les autres lettres. En typographie professionnelle, le point occupe très peu de place et il est placé tout près du caractère qui le précède. Il n'est donc pas possible d'utiliser trois points consécutifs pour créer des points de suspension. À la place on utilise la commande spécifique :

`\dots`

```
Non pas comme \c{c}a...
mais ainsi :\\
New York, Tokyo, Budapest\dots
```

Non pas comme ça... mais ainsi :
New York, Tokyo, Budapest...

2.4.6 Ligatures

Certaines séquences de lettres ne sont pas composées simplement en juxtaposant les différentes lettres les unes à la suite des autres, mais en utilisant des symboles spéciaux.

ff fi fl ffi... à la place de ff fi fl ffi...

Ces ligatures peuvent être désactivées en insérant un `\mbox{}` entre les lettres en question. Cela peut s'avérer utile pour certains mots composés[6].

```
Not shelfful\\
but shelf\mbox{}ful
```

Not shelfful
but shelfful

[6] Il n'existe pas d'exemple en français. *NdT.*

2.4.7 Accents et caractères spéciaux

LaTeX permet l'utilisation d'accents et de caractères spéciaux issus de nombreuses langues. Le tableau 2.1 montre tous les accents que l'on peut ajouter à la lettre o. Ils s'appliquent naturellement aux autres lettres.

Pour placer un accent sur un i ou un j, il faut supprimer leur point. Ceci s'obtient en tapant `\i` et `\j`.

```
H\^otel, na\"\i ve, \'el\‘eve,\\
sm\o rrebr\o d, !‘Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

Tab. 2.1 – Accents et caractères spéciaux

ò	`\‘o`	ó	`\'o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`	ç	`\c c`
ǒ	`\u o`	ǒ	`\v o`	ő	`\H o`	ǫ	`\c o`
ọ	`\d o`	o͟	`\b o`	o͡o	`\t oo`		
œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`
å	`\aa`	Å	`\AA`				
ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ı	`\i`	ȷ	`\j`	¡	`!‘`	¿	`?‘`

2.5 Support multilingue

Pour composer des documents dans des langues autres que l'anglais, LaTeX doit pouvoir s'adapter aux règles typographiques et aux règles de césure propres à chaque langue. Il y a plusieurs domaines pour lesquels il faut configurer LaTeX pour chaque langue :

1. Toutes les chaînes de caractères générées automatiquement [7] doivent être traduites. Pour de nombreuses langues, ces adaptations peuvent être réalisées par l'extension babel de Johannes Braams.
2. LaTeX doit connaître les règles de césure de la nouvelle langue. Définir les règles de césure utilisées par LaTeX est une tâche assez complexe, qui impose la construction de formats spécifiques. Votre *Local Guide* [5] devrait vous indiquer quelles sont les langues supportées par votre installation de LaTeX et comment configurer les règles de césure.

[7] « Table des matières », « Liste des figures »,...

3. Certaines règles typographiques changent en fonction de la langue ou de la région géographique. Ces changements peuvent être supportés par l'extension babel ou par des extensions spécifiques (telles que french [12] pour le français).

Si votre système est configuré correctement, vous pouvez sélectionner la langue utilisée par l'extension babel avec la commande :

```
\usepackage[langue]{babel}
```

après la commande `\documentclass`. Les *langue*s supportées par votre système sont normalement listées dans votre *Local Guide* [5]. Lorsque la césure d'une langue n'est pas supportée par votre moteur LaTeX, celle-ci peut être utilisée avec l'extension babel mais à cause des césures incorrectes le résultat ne sera pas satisfaisant.

Pour certaines langues babel définit également de nouvelles commandes qui simplifient la saisie des caractères spéciaux.

Les systèmes informatiques modernes vous permettent de saisir directement les caractères accentués ou les symboles spécifiques d'une langue. Depuis la version de décembre 1994, LaTeX 2_ε sait gérer ce type de codage grâce à l'extension inputenc :

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

Les codages suivants sont utilisés en fonction du système d'exploitation pour les langues d'Europe de l'Ouest :

Système	Codage
Mac OS	`applemac`
Unix	`latin1`
Windows	`ansinew`
OS/2	`cp850`

Pour faciliter la lecture, cette extension est utilisée dans la suite du document pour représenter les caractères accentués dans les exemples.

En utilisant cette extension, il faut prendre garde au fait que les systèmes informatiques n'utilisent pas tous le même codage des caractères spéciaux. Le caractère *é* par exemple, est codé 142 sur Mac et 233 sur PC Windows ou sous Unix (codage latin-1). D'autres personnes risquent donc de ne pas pouvoir lire votre document source, parce que leur système utilise un codage différent.

Lorsqu'un document source LaTeX doit être transmis vers un autre système, il est recommandé de convertir les caractères accentués et spéciaux en séquences du type `\'e` préalablement au transfert. Il existe de nombreux utilitaires pour faire la conversion automatique dans les deux sens : `recode` sous Unix, `Tower of Babel` sur Macintosh, etc.

Le codage des polices de caractères est une autre histoire. Celui-ci définit à quelle position dans une police de TEX se trouve chaque caractère. La police originale de TEX, Computer Modern, ne contient que les 128 caractères du jeu de caractères ASCII. Pour produire un caractère accentué, TEX combine un caractère normal avec un accent. Bien que le résultat obtenu ainsi soit presque parfait, cette approche empêche la césure automatique des mots contenant des accents.

Heureusement, dans la plupart des installations modernes de TEX on trouve la police EC. Cette police à la même apparence que la police Computer Modern, mais elle contient également les caractères accentués pour la plupart des langues européeenes. L'utilisation de cette police permet donc la césure des mots contenant des caractères accentués. L'utilisation de la police EC est activée par l'extension fontenc, avec la commande suivante dans le préambule :

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

2.5.1 Support de la langue française

Pour le français, l'option francais de l'extension babel applique les règles de césure spécifiques du français et adapte LATEX à la plupart des règles spécifiques de la typographie française [13] : présentation des listes, insertion automatique de l'espacement avant les signes de ponctuation doubles, etc.

```
\usepackage[francais]{babel}
```

TAB. 2.2 – Commandes de saisie en français.

`\og guillemets \fg{}`	« guillemets »
`M\up{me}, D\up{r}`	M^{me}, D^{r}
`1\ier{}, 1\iere{}, 1\ieres{}`	1^{er}, 1^{re}, 1^{res}
`2\ieme{} 4\iemes{}`	2^{e} 4^{es}
`\No 1, \no 2`	N^{o} 1, n^{o} 2
`20~\degres C, 45\degres`	20 °C, 45°
`\bsc{M. Durand}`	M. DURAND
`\nombre{1234,56789}`	1 234,567 89

Les mots générés automatiquement par LATEX sont traduits en français et certaines commandes supplémentaires (cf. table 2.2) sont disponibles.

Dans cette traduction, un certain nombre d'ajouts présentent ces spécificités de la typographie française tout au long du texte.

2.5.2 Support de la langue allemande

Voici quelques conseils pour créer des documents en allemand à l'aide de LaTeX. Le support de la langue allemande est activé par la commande :

```
\usepackage[german]{babel}
```

La césure allemande est alors activée, si votre système a été configuré pour cela. Le texte produit automatiquement par LaTeX est traduit en allemand (par ex. "Kapitel" pour un chapitre). De nouvelles commandes (listées dans la table 2.3) permettent la saisie simplifiée des caractères spéciaux.

Tab. 2.3 – Caractères spéciaux en allemand.

`"a`	ä	`"s`	ß	`"‘`	„
`"’`	“	`"<`	«	`">`	»
`\dq`	"				

2.6 L'espace entre les mots

Pour obtenir une marge droite alignée, LaTeX insère des espaces plus ou moins larges entre les mots.

Un tilde « `~` » produit une espace interdisant tout saut de ligne (dit espace *insécable*). `~` est à utiliser pour éviter les coupures indésirables : on code `M.~Dupont`, `cf.~Fig.~5`, etc.

La commande `\,` permet d'insérer une demi-espace insécable.

```
1\,234\,567
```

1 234 567

Signalons enfin une subtilité : après la ponctuation finale d'une phrase, les règles de la typographie anglo-saxonne veulent que l'on insère une espace plus large. Mais si un point suit une lettre majuscule, LaTeX considère qu'il s'agit d'une abréviation et insère alors une espace normale. La commande `\@` avant un point indique que celui-ci termine une phrase, même lorsqu'il suit une lettre majuscule.

```
M.~Dupont vous remercie\\
R.E.M.\\
I like BASIC\@. Do you?
```

M. Dupont vous remercie
R.E.M.
I like BASIC. Do you?

L'ajout d'espace supplémentaire à la fin d'une phrase peut être supprimé par la commande :

`\frenchspacing`

qui est active par défaut avec l'option francais de l'extension babel. Dans ce cas, la commande `\@` n'est pas nécessaire.

2.7 Titres, chapitres et sections

Pour aider le lecteur à suivre votre pensée, vous souhaitez séparer vos documents en chapitres, sections ou sous-sections. LaTeX utilise pour cela des commandes qui prennent en argument le titre de chaque élément. C'est à vous de les utiliser dans l'ordre.

Dans la classe de document `article`, les commandes de sectionnement suivantes sont disponibles :

```
\part{...}
\section{...}              \paragraph{...}
\subsection{...}           \subparagraph{...}
\subsubsection{...}        \appendix
```

Dans les classes `report` et `book`, la commande

```
\chapter{...}
```

est également reconnue, (elle s'intercale entre `\part` et `\section`)[8].

L'espacement entre les sections, la numérotation et le choix de la police et de la taille des titres sont gérés automatiquement par LaTeX.

Deux commandes de sectionnement ont un comportement spécial :

- la commande `\part` ne change pas la numérotation des chapitres ;
- la commande `\appendix` ne prend pas d'argument. Elle bascule simplement la numérotation des chapitres[9] en lettres.

LaTeX peut ensuite créer la table des matières en récupérant la liste des titres et de leur numéro de page d'une exécution précédente (fichier `.toc`).

[8] Puisque la classe `article` ne connaît pas les chapitres, il est facile par exemple de regrouper des articles en tant que chapitres d'un livre en remplacant le `\title` de chaque article par `\chapter`.

[9] Pour la classe article, elle change la numérotation des sections

La commande :

`\tableofcontents`

imprime la table des matières. Un document doit être traité (on dit aussi « compilé ») deux fois par LaTeX pour avoir une table des matières correcte. Dans certains cas, un troisième passage est même nécessaire. LaTeX vous indique quand cela est nécessaire.

Toutes les commandes citées ci-dessus existent dans une forme « étoilée » obtenue en ajoutant une étoile `*` au nom de la commande. Ces commandes produisent des titres de sections qui n'apparaissent pas dans la table des matières et qui ne sont pas numérotés. On peut ainsi remplacer la commande `\section{Introduction}` par `\section*{Introduction}`.

Par défaut, les titres de section apparaissent dans la table des matières exactement comme ils sont dans le texte. Parfois il n'est pas possible de faire tenir un titre trop long dans la table des matières. On peut donner un titre spécifique pour la table des matières en argument optionnel avant le titre principal :

```
\chapter[Le LAAS du CNRS]{Le Laboratoire
         d'Analyse et d'Architecture
        des Systèmes du Centre National
        de la Recherche Scientifique}
```

Le titre du document est obtenu par la commande :

`\maketitle`

Les éléments de ce titre sont définis par les commandes :

`\title{...}`, `\author{...}` et éventuellement `\date{...}`

qui doivent être appelées avant `\maketitle`. Dans l'argument de la commande `\author`, vous pouvez citer plusieurs auteurs en séparant leurs noms par des commandes `\and`.

Vous trouverez un exemple des commandes citées ci-dessus sur la figure 1.3, page 9.

En plus des commandes de sectionnement expliquées ci-dessus, LaTeX 2_ε a introduit trois nouvelles commandes destinées à être utilisées avec la classe `book` :

`\frontmatter` doit être la première commande après le `\begin{document}`, elle introduit le préambule du document. Les numéros de pages sont alors en romain (i, ii, iii, etc.). En général on ne numérote pas les parties d'un préambule, ceci se fait en utilisant les variantes étoilées des commandes de sectionnement comme `\chapter*{Preface}`.

`\mainmatter` se place juste avant le début du premier (vrai) chapitre du livre, la numérotation des pages se fait alors en chiffres arabes et le compteur de pages est remis à 1.

`\appendix` indique le début des appendices, les numéros des chapitres sont alors remplacés par des lettres majuscules (A, B, etc.).

`\backmatter` se place juste avant la bibliographie et les index.

2.8 Références croisées

Dans les livres, rapports ou articles, on trouve souvent des références croisées vers des figures, des tableaux ou des passages particuliers du texte. LaTeX dispose des commandes suivantes pour faire des références croisées :

`\label{`*marque*`}`, `\ref{`*marque*`}` et `\pageref{`*marque*`}`

où *marque* est un identificateur choisi par l'utilisateur. LaTeX remplace `\ref` par le numéro de la section, de la sous-section, de la figure, du tableau, ou du théorème où la commande `\label` correspondante a été placée. L'utilisation de références croisées rend nécessaire de compiler deux fois le document : à la première compilation les numéros correspondant aux étiquettes `\label{}` sont inscrits dans le fichier `.aux` et, à la compilation suivante, `\ref{}` et `\pageref{}` peuvent imprimer ces numéros[10].

```
Une référence à cette
section\label{ma-section}
ressemble à :
«~voir section~\ref{ma-section},
page~\pageref{ma-section}.~»
```

Une référence à cette section ressemble à : « voir section 20, page 28. »

2.9 Notes de bas de page

La commande :

`\footnote{`*texte*`}`

imprime une note de bas de page en bas de la page en cours. Les notes de bas de page doivent être placées après le mot où la phrase auquel elles se réfèrent[11].

[10] Ces commandes ne connaissent pas le type du numéro auquel elles se réfèrent, elles utilisent le dernier numéro généré automatiquement.

[11] La typographie française demande une espace fine avant la marque de renvoi à la note.

```
Les notes de bas de page%
\,\footnote{ceci est une note
  de bas de page.}
sont très prisées par les
utilisateurs de \LaTeX{}.
```

Les notes de bas de page[a] sont très prisées par les utilisateurs de LaTeX.

[a] ceci est une note de bas de page.

2.10 Souligner l'importance d'un mot

Dans un manuscrit réalisé sur une machine à écrire, les mots importants sont soulignés, ce qui peut être obtenu par la commande:

`\underline{`*texte*`}`

Dans un ouvrage imprimé, on préfère les *mettre en valeur* [12]. La commande de mise en valeur est :

`\emph{`*texte*`}`

Son argument est le texte à mettre en valeur. En général, la police *italique* est utilisée pour la mise en valeur, sauf si le texte est déja en italique, auquel cas on utilise une police romaine (droite).

```
\emph{Pour \emph{insister}
dans un passage déjà
mis en valeur, \LaTeX{}
utilise une police droite.}
```

Pour insister *dans un passage déjà mis en valeur, LaTeX utilise une police droite.*

Remarquez la différence entre demander à LaTeX de *mettre en valeur* un mot et lui demander de changer de *police de caractères* :

```
\textit{Vous pouvez aussi
  \emph{mettre en valeur} du
 texte en italique,}
\textsf{ou dans une police
  \emph{sans-serif},}
\texttt{ou dans une police
  \emph{machine à écrire}.}
```

Vous pouvez aussi mettre en valeur *du texte en italique,* ou dans une police *sans-serif*, ou dans une police *machine à écrire*.

[12] *Emphasized* en anglais.

2.11 Environnements

Pour composer du texte dans des contextes spécifiques, LaTeX définit des environnements différents pour divers types de mise en page :

`\begin{`*nom*`}` *texte* `\end{`*nom*`}`

nom est le nom de l'environnement. Les environnements peuvent être imbriqués .

`\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}`

Dans les sections suivantes tous les environnements importants sont présentés.

2.11.1 Listes, énumérations et descriptions

L'environnement `itemize` est utilisé pour des listes simples, `enumerate` est utilisé pour des énumérations (listes numérotées) et `description` est utilisé pour des descriptions.

```
\begin{enumerate}
\item Il est possible d'imbriquer
les environnements à sa guise :
\begin{itemize}
\item mais cela peut ne pas
  être  très beau,
\item ni facile à suivre.
\end{itemize}
\item Souvenez-vous :
\begin{description}
\item[Clarté :] les faits ne
vont pas devenir plus sensés
parce  qu'ils sont dans une liste,
\item[Synthèse :] cependant une
liste peut très bien
résumer des faits.
\end{description}
\end{enumerate}
```

1. Il est possible d'imbriquer les environnements à sa guise :
   - mais cela peut ne pas être très beau,
   - ni facile à suivre.
2. Souvenez-vous :

   Clarté : les faits ne vont pas devenir plus sensés parce qu'ils sont dans une liste,

   Synthèse : cependant une liste peut très bien résumer des faits.

Notez que l'option francais de l'extension babel utilise une présentation des listes simples qui respecte les règles typographiques françaises :

```
Une liste simple française :
\begin{itemize}
\item voici un élément ;
\item puis un autre.
\end{itemize}
```

Une liste simple française :
– voici un élément ;
– puis un autre.

```
An english list:
\begin{itemize}
\item one item
\item an other one
\end{itemize}
```

An english list:

- one item
- an other one

2.11.2 Alignements à gauche, à droite et centrage

Les environnements `flushleft` et `flushright` produisent des textes alignés à gauche ou à droite. L'environnement `center` produit un texte centré. Si vous n'utilisez pas la commande \\ pour indiquer les sauts de ligne, ceux-ci sont calculés automatiquement par LaTeX.

```
\begin{flushleft}
Ce texte est\\
aligné à gauche.
\LaTeX{} n'essaye pas
d'aligner la marge droite.
\end{flushleft}
```

Ce texte est
aligné à gauche. LaTeX n'essaye pas d'aligner la marge droite.

```
\begin{flushright}
Ce texte est\\
aligné à droite.
\LaTeX{} n'essaye pas
d'aligner la marge gauche.
\end{flushright}
```

Ce texte est
aligné à droite. LaTeX n'essaye pas d'aligner la marge gauche.

```
\begin{center}
Au centre de la terre.
\end{center}
```

Au centre de la terre.

2.11.3 Citations et vers

L'environnement `quote` est utile pour les citations, les phrases importantes ou les exemples.

```
Une règle typographique
simple pour la longueur
des lignes :
\begin{quote}
Une ligne ne devrait pas comporter
plus de 66~caractères.

C'est pourquoi les pages
composées par \LaTeX{} ont des
marges importantes.
\end{quote}
Cela explique pourquoi les
journaux utilisent souvent
plusieurs colonnes.
```

Une règle typographique simple pour la longueur des lignes :

> Une ligne ne devrait pas comporter plus de 66 caractères.
>
> C'est pourquoi les pages composées par LaTeX ont des marges importantes.

Cela explique pourquoi les journaux utilisent souvent plusieurs colonnes.

Il existe deux autres environnements comparables : `quotation` et `verse`. L'environnement `quotation` est utile pour des citations plus longues, couvrant plusieurs paragraphes parce qu'il indente les paragraphes. L'environnement `verse` est utilisé pour la poésie, là où les retours à la ligne sont importants. Les vers sont séparés par des commandes `\\` et les strophes par une ligne vide[13].

```
Voici le début d'une
fugue de Boris Vian :
\begin{flushleft}
\begin{verse}
Les gens qui n'ont plus
  rien à faire\\
Se suivent dans la rue comme\\
Des wagons de chemin de fer.

Fer fer fer\\
Fer fer fer\\
Fer quoi faire\\
Fer coiffeur.\\
\end{verse}
\end{flushleft}
```

Voici le début d'une fugue de Boris Vian :

> Les gens qui n'ont plus rien à faire
> Se suivent dans la rue comme
> Des wagons de chemin de fer.
>
> Fer fer fer
> Fer fer fer
> Fer quoi faire
> Fer coiffeur.

2.11.4 Impression *verbatim*

Tout texte inclus entre `\begin{verbatim}` et `\end{verbatim}` est imprimé tel quel, comme s'il avait été tapé à la machine, avec tous les retours à la ligne et les espaces, sans qu'aucune commande LaTeX ne soit exécutée.

[13] Les puristes constateront que l'environnement `verse` ne respecte pas les règles de la typographie française : les rejets devraient être préfixés par / et alignés à droite sur le vers précédent.

À l'intérieur d'un paragraphe, une fonctionnalité équivalente peut être obtenue par

`\verb+`*texte*`+`

Le caractère `+` est seulement un exemple de caractère séparateur. Vous pouvez utiliser n'importe quel caractère, sauf les lettres, `*` ou l'espace. La plupart des exemples de commandes LaTeX dans ce document sont réalisés avec cette commande.

```
La commande \verb|\dots| \dots

\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

La commande \dots ...

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
La version étoilée de
l'environnement  verbatim
met    les    espaces    en
évidence
\end{verbatim*}
```

```
La␣version␣étoilée␣de
l'environnement␣␣verbatim
met␣␣␣␣les␣␣␣espaces␣␣␣en
évidence
```

La commande `\verb` peut également être utilisée avec une étoile :

```
\verb*|comme ceci :-) |
```

```
comme␣ceci␣:-)␣
```

L'environnement `verbatim` et la commande `\verb` ne peuvent être utilisés à l'intérieur d'autres commandes comme `\footnote{}`.

2.11.5 Tableaux

L'environnement `tabular` permet de réaliser des tableaux avec ou sans lignes de séparation horizontales ou verticales. LaTeX ajuste automatiquement la largeur des colonnes.

L'argument *description du tableau* de la commande :

`\begin{tabular}{`*description du tableau*`}`

définit le format des colonnes du tableau. Utilisez un `l` pour une colonne alignée à gauche, `r` pour une colonne alignée à droite et `c` pour une colonne centrée. `p{`*largeur*`}` permet de réaliser une colonne justifiée à droite sur plusieurs lignes et enfin `|` permet d'obtenir une ligne verticale.

À l'intérieur de l'environnement `tabular`, le caractère `&` est le séparateur de colonnes, `\\` commence une nouvelle ligne et `\hline` insère une ligne horizontale.

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadécimal \\
3700 & octal \\
11111000000 & binaire \\
\hline \hline
1984 & décimal \\
\hline
\end{tabular}
```

7C0	hexadécimal
3700	octal
11111000000	binaire
1984	décimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Bienvenue dans ce
cadre.\\
Merci de votre visite.\\
\hline
\end{tabular}
```

Bienvenue dans ce cadre. Merci de votre visite.

La construction `@{...}` permet d'imposer le séparateur de colonnes. Cette commande supprime l'espacement inter-colonnes et le remplace par ce qui est indiqué entre les crochets. Une utilisation courante de cette commande est présentée plus loin comme solution au problème de l'alignement des nombres décimaux. Une autre utilisation possible est de supprimer l'espacement dans un tableau avec `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
sans espace\\\hline
\end{tabular}
```

sans espace

```
\begin{tabular}{l}
\hline
avec espaces\\
\hline
\end{tabular}
```

avec espaces

S'il n'y a pas de commande prévue [14] pour aligner les nombres sur le point décimal (ou la virgule si on respecte les règles françaises) nous pouvons « tricher » et réaliser cet alignement en utilisant deux colonnes : la

[14] Si les extensions de l'ensemble « tools » sont installées sur votre système, jetez un œil sur l'extension dcolumn faite pour résoudre ce problème.

première alignée à droite contient la partie entière et la seconde alignée à gauche contient la partie décimale. La commande `\@{,}` dans la description du tableau remplace l'espace normale entre les colonnes par une simple virgule, donnant l'impression d'une seule colonne alignée sur le séparateur décimal. N'oubliez pas de remplacer dans votre tableau le point ou la virgule par un séparateur de colonnes (`&`) ! Un label peut être placé au-dessus de cette colonne en utilisant la commande `\multicolumn`.

```
\begin{tabular}{c r @{,} l}
Expression       &
\multicolumn{2}{c}{Valeur} \\
\hline
$\pi$                & 3&1416  \\
$\pi^{\pi}$          & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

Expression	Valeur
π	3,1416
π^{π}	36,46
$(\pi^{\pi})^{\pi}$	80662,7

Autre exemple d'utilisation de `\multicolumn` :

```
\begin{tabular}{|l|l|}
\hline
\multicolumn{2}{|c|}{\textbf{Nom}}\\
\hline
Dupont & Jules \\
\hline
\end{tabular}
```

Nom	
Dupont	Jules

LaTeX traite le contenu d'un environnement `tabular` comme une boîte indivisible, en particulier il ne peut y avoir de coupure de page. Pour réaliser de longs tableaux s'étendant sur plusieurs pages il faut avoir recours aux extensions supertabular ou longtabular.

2.12 Objets flottants

De nos jours, la plupart des publications contiennent un nombre important de figures et de tableaux. Ces éléments nécessitent un traitement particulier car ils ne peuvent être coupés par un changement de page. On pourrait imaginer de commencer une nouvelle page chaque fois qu'une figure ou un tableau ne rentrerait pas dans la page en cours. Cette façon de faire laisserait de nombreuses pages à moitié blanches, ce qui ne serait réellement pas beau.

La solution est de laisser « flotter » les figures et les tableaux qui ne rentrent pas sur la page en cours, vers une page suivante et de compléter la page avec le texte qui suit l'objet « flottant ». LaTeX fournit deux environnements pour les objets flottants adaptés respectivement aux figures (`figure`)

et aux tableaux (`table`). Pour faire le meilleur usage de ces deux environnements, il est important de comprendre comment LaTeX traite ces objets flottants de manière interne. Dans le cas contraire ces objets deviendront une cause de frustration intense car LaTeX ne les placera jamais à l'endroit où vous souhaitiez les voir.

Commençons par regarder les commandes que LaTeX propose pour les objets flottants :

Tout objet inclus dans un environnement `figure` ou `table` est traité comme un objet flottant. Les deux environnements flottants ont un paramètre optionnel :

`\begin{figure}[`*placement*`]` ou `\begin{table}[`*placement*`]`

appelé *placement*. Ce paramètre permet de dire à LaTeX où vous autorisez l'objet à flotter. Un *placement* est composé d'une chaîne de caractères représentant des *placements possibles*. Reportez-vous au tableau 2.4.

Tab. 2.4 – Placements possibles

Caractère	Emplacement pour l'objet flottant...
`h`	*here*, ici, à l'emplacement dans le texte où la commande se trouve. Utile pour les petits objets.
`t`	*top*, en haut d'une page
`b`	*bottom*, en bas d'une page
`p`	*page*, sur une page à part ne contenant que des objets flottants.
`!`	ici, sans prendre en compte les paramètres internes[a] qui pourraient empêcher ce placement.

[a] tels que le nombre maximum d'objets flottants sur une page

Un tableau peut commencer par exemple par la ligne suivante :

```
\begin{table}[!hbp]
```

L'emplacement `[!hbp]` permet à LaTeX de placer le tableau soit sur place (`h`), soit en bas de page (`b`) soit enfin sur une page à part (`p`), et tout cela même si les règles internes de LaTeX ne sont pas toutes respectées (`!`). Si aucun placement n'est indiqué, les classes standard utilisent `[tbp]` par défaut.

LaTeX place tous les objets flottants qu'il rencontre dans l'ordre en suivant les indications fournies par l'auteur. Si un objet ne peut être placé sur la page en cours, il est placé soit dans la file des figures soit dans la file des tableaux[15]. Quand une nouvelle page est entamée, LaTeX essaye d'abord de

[15] Il s'agit de files FIFO (*First In, First Out*) : premier arrivé, premier servi.

voir si les objets en tête des deux files pourraient être placés sur une page spéciale, à part. Si cela n'est pas possible, les objets en tête des deux files sont traités comme s'ils venaient d'être trouvés dans le texte : LaTeX essaye de les placer selon les placements possibles restants. Tous les nouveaux objets flottants rencontrés dans la suite du texte sont ajoutés à la queue des files. LaTeX respecte scrupuleusement l'ordre d'apparition des objets flottants. C'est pourquoi un objet flottant qui ne peut être placé dans le texte repousse tous les autres à la fin du document.

D'où la règle :

> Si LaTeX ne place pas les objets flottants comme vous le souhaitez, c'est souvent à cause d'un seul objet trop grand qui bouche l'une des deux files d'objets flottants.

Essayer d'imposer à LaTeX un emplacement particulier en utilisant l'option `[h]` pour un flottant est une idée *à proscrire*, les versions modernes de LaTeX changent d'ailleurs automatiquement l'option `[h]` en `[ht]`.

Voici quelques éléments supplémentaires qu'il est bon de connaître sur les environnements `table` et `figure`.

Avec la commande :

`\caption{`*texte de la légende*`}`

vous définissez une légende pour l'objet. Un numéro (incrémenté automatiquement) et le mot « Figure » ou « Table »[16] sont ajoutés par LaTeX.

Les deux commandes :

`\listoffigures` et `\listoftables`

fonctionnent de la même manière que la commande `\tableofcontents` ; elles impriment respectivement la liste des figures et des tableaux. Dans ces listes, la légende est reprise en entier. Si vous désirez utiliser des légendes longues, vous pouvez en donner une version courte entre crochets qui sera utilisée pour la table :

```
\caption[courte]{LLLLLooooooonnnnnggggguuuueee}
```

Avec `\label` et `\ref` vous pouvez faire référence à votre objet à l'intérieur de votre texte.

L'exemple suivant dessine un carré et l'insère dans le document. Vous pouvez utiliser cette commande pour réserver de la place pour une illustration que vous allez coller sur le document terminé.

```
La figure~\ref{blanche} est un exemple de Pop-Art.
\begin{figure}[!hbp]
```

[16] Avec l'extension `babel`, la présentation des légendes est modifiée pour obéir aux règles françaises.

```
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Cinq centimètres sur cinq} \label{blanche}
\end{figure}
```

Dans l'exemple ci-dessus[17] LaTeX va s'acharner (`!`) à placer la figure là où se trouve la commande (`h`) dans le texte. S'il n'y arrive pas, il essayera de la placer en bas (`b`) de la page. Enfin s'il ne peut la placer sur la page courante, il essayera de créer une page à part avec d'autres objets flottants. S'il n'y a pas suffisamment de tableaux en attente pour remplir une page spécifique, LaTeX continue et, au début de la page suivante, réessayera de placer la figure comme si elle venait d'apparaître dans le texte.

Dans certains cas il peut s'avérer nécessaire d'utiliser la commande :

`\clearpage` ou même `\cleardoublepage`

Elle ordonne à LaTeX de placer tous les objets en attente immédiatement puis de commencer une nouvelle page. `\cleardoublepage` commence une nouvelle page de droite.

À la section 4.1, vous apprendrez à inclure une figure PostScript dans vos documents.

2.13 Protection des commandes « fragiles »

Les arguments de commandes telles que `\section` ou `\caption` etc., peuvent apparaître plusieurs fois dans le document (par exemple aussi dans la table des matières, les hauts de pages...), on dit qu'il s'agit d'arguments « mobiles » (*moving arguments*). Certaines commandes, entre autres `\footnote`, `\phantom` etc., ne produisent pas le résultat escompté quand elles sont exécutées comme argument de commandes de type `\section`, on dit qu'elles sont « fragiles », ce qui signifie qu'elles ont besoin de la protection d'un ... `\protect`.

La commande `\protect` n'a d'effet que sur la commande qui la suit immédiatement, mais *pas ses arguments* éventuels. La plupart du temps un `\protect` de trop ne produira aucun effet pervers.

Voici un exemple d'utilisation de `\protect` :

```
\section{Je suis prudent
   \protect\footnote{Je protège ma note de bas de page.}}
```

[17] En supposant que la file des figures soit vide.

Chapitre 3

Formules Mathématiques

Vous êtes prêts ! Dans ce chapitre nous allons aborder l'atout majeur de TEX : la composition de formules mathématiques. Mais attention, ce chapitre ne fait que décrire les commandes de base. Bien que ce qui est expliqué ici soit suffisant pour la majorité des utilisateurs, ne désespérez pas si vous n'y trouvez pas la solution à votre problème de mise en forme d'une équation mathématique. Il y a de fortes chances pour que la solution se trouve dans l'extension amsmath de $\mathcal{AMS}$-LATEX [1].

3.1 Généralités

LATEX dispose d'un mode spécial pour la mise en page de formules mathématiques. Des maths à l'intérieur d'un paragraphe sont saisies entre `\(` et `\)`, entre `$` et `$` ou entre `\begin{math}` et `\end{math}`.

```
Ajoutez $a$ au carré
et $b$ au carré pour obtenir
$c$ au carré. Ou, en
utilisant une approche plus
matheuse : $c^{2}=a^{2}+b^{2}$
```

Ajoutez a au carré et b au carré pour obtenir c au carré. Ou, en utilisant une approche plus matheuse : $c^2 = a^2 + b^2$

```
100~m$^{3}$ d'eau\\[6pt]
J'$\heartsuit$ \LaTeX{}
```

100 m^3 d'eau

J'♡ LATEX

Il vaut mieux composer les équations ou les formules plus importantes « *hors-texte* », c'est-à-dire sur des lignes à part. Pour cela, on les inclut entre `\[` et `\]` ou entre `\begin{displaymath}` et `\end{displaymath}`. On obtient

[1] *American Mathematical Society* = Société Américaine de Mathématiques.

ainsi des formules qui ne sont pas numérotées. Si vous voulez qu'elles soient numérotées par LaTeX, utilisez l'environnement `equation`.

```
Ajoutez $a$ au carré
et $b$ au carré pour obtenir
$c$ au carré. Ou, en
utilisant une approche plus
matheuse :
\[ c^{2}=a^{2}+b^{2} \]
Le mot de la fin.
```

Ajoutez a au carré et b au carré pour obtenir c au carré. Ou, en utilisant une approche plus matheuse :

$$c^2 = a^2 + b^2$$

Le mot de la fin.

Avec `\label` et `\ref`, vous pouvez faire référence à une équation.

```
\begin{equation} \label{eq}
\epsilon > 0
\end{equation}
L'équation (\ref{eq})
nous donne\dots
```

$$\epsilon > 0 \tag{3.1}$$

L'équation (3.1) nous donne...

Remarquez que les expressions mathématiques sont formatées différemment selon qu'elles sont composées « en ligne » ou « hors texte ». Comparez :

```
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Il y a des différences entre le mode *mathématique* et le mode *texte*. Par exemple, en mode *mathématique* :

1. la plupart des espaces et des retours à la ligne n'ont aucune signification. Les espaces sont déduites de la logique de la formule ou indiquées à l'aide de commandes spécifiques telles que : `\,`, `\quad` ou`\qquad`;
2. les lignes vides ne sont pas autorisées. Un seul paragraphe par formule ;
3. chaque lettre est considérée comme étant le nom d'une variable et sera imprimée comme tel. Pour insérer du texte normal (police et espacement standard) dans une formule, il faut utiliser la commande `\textrm{...}`.

```
\begin{equation}
\forall x \in \mathbf{R}:
\qquad x^{2} \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R} : \qquad x^2 \geq 0 \tag{3.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{pour tout }
  x\in\mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \qquad \text{pour tout } x \in \mathbf{R} \tag{3.3}$$

Une mode récente et contestable pousse à utiliser la police « blackboard bold » (Gras Tableau Noir, ainsi appelée car c'est par le doublement des verticales des lettres que l'on simule le gras typographique lorsqu'on ne peut faire autrement) qui est obtenue par la commande `\mathbb` de l'extension amsfonts ou amssymb pour désigner les ensembles de nombres entiers, réels, etc. L'exemple précédent devient :

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{pour tout }x\in\mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \qquad \text{pour tout } x \in \mathbb{R}$$

3.2 Groupements en mode mathématique

La plupart des commandes du mode mathématique ne s'applique qu'au caractère suivant. Pour qu'une commande s'applique à un ensemble de caractères, il faut les grouper en utilisant des accolades : `{...}`.

```
\begin{equation}
a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \tag{3.4}$$

3.3 Éléments d'une formule mathématique

Dans cette section nous allons voir les commandes les plus importantes du mode mathématique. Pour une liste de tous les symboles disponibles, voyez la section 3.11, page 53.

Les lettres **grecques minuscules** sont saisies de la manière suivante : `\alpha`, `\beta`, `\gamma`, etc. Les lettres **grecques majuscules** [2] sont saisies ainsi : `\Gamma`, `\Delta`, etc.

[2] Il n'y a pas de Alpha majuscule dans LaTeX 2ε parce que c'est le même caractère que le A romain. Lorsque le nouveau codage mathématique sera terminé, cela changera.

```
$\lambda,\xi,\pi,\mu,\Phi,\Omega$
```

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Les **indices et exposants** sont positionnés en utilisant les caractères `_` et `^`.

```
$a_{1}$ \qquad $x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$\\
$e^{x^2} \neq ({e^x})^2$
```

$a_1 \qquad x^2 \qquad e^{-\alpha t} \qquad a^3_{ij}$
$e^{x^2} \neq (e^x)^2$

La **racine carrée** est saisie ainsi : `\sqrt`, la racine $n^{\text{ième}}$ est produite par la commande `\sqrt[`n`]`. La taille du symbole racine est calculée par LaTeX. Pour obtenir le symbole seul, utilisez `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

$\sqrt{x} \qquad \sqrt{x^2+\sqrt{y}} \qquad \sqrt[3]{2}$
$\surd[x^2 + y^2]$

Les commandes `\overline` et `\underline` créent un **trait horizontal** au-dessus ou au-dessous d'une expression.

```
$\overline{m+n}$
```

$\overline{m+n}$

Les commandes `\overbrace` et `\underbrace` créent une grande **accolade horizontale** au-dessus ou au-dessous d'une expression.

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

$\underbrace{a+b+\cdots+z}_{26}$

Pour ajouter des accents mathématiques tels que des flèches ou des tildes, vous pouvez utiliser les commandes du tableau 3.1 p. 53. Les chapeaux et les tildes larges, couvrant plusieurs caractères, sont produits par les commandes `\widetilde` et `\widehat`. La commande `'` produit un prime.

```
\begin{displaymath}
y=x^{2}\qquad y'=2x\qquad y''=2
\end{displaymath}
```

$$y = x^2 \qquad y' = 2x \qquad y'' = 2$$

Les **vecteurs** sont en général marqués en ajoutant une flèche au-dessus du nom de la variable. Ceci est obtenu par la commande `\vec`. Pour coder

le vecteur de A à B, les commandes `\overrightarrow` et `\overleftarrow` sont bien utiles.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec{a} \quad \overrightarrow{AB}$$

En général, les points indiquant une opération de multiplication ne sont pas imprimés. Cependant, il arrive qu'il soit nécessaire de les faire apparaître pour aider la lecture. Utilisez alors `\cdot`.

```
\[ v = \sigma{}_1 \cdot \sigma{}_2
  \tau{}_1 \cdot \tau{}_2 \]
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

Les noms des fonctions telles que sinus doivent être imprimés à l'aide d'une police droite et non en italique comme les variables. LaTeX fournit donc les commandes suivantes pour les fonctions les plus utilisées :

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

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \to 0} \frac{\sin x}{x} = 1$$

Pour la fonction modulo, il y a deux commandes possibles : `\bmod` pour l'opérateur binaire et `\pmod` pour l'opérateur unaire :

```
$a\bmod b$\\
$x\equiv a \pmod{b}$
```

$a \bmod b$
$x \equiv a \pmod{b}$

Un trait de **fraction** est produit par la commande :

`\frac{`*numérateur*`}{`*dénominateur*`}`

La forme utilisant un *slash* (1/2) est souvent préférable pour des petits éléments.

```
$1\frac{1}{2}$~hours
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

$1\frac{1}{2}$ hours

$$\frac{x^2}{k+1} \qquad x^{\frac{2}{k+1}} \qquad x^{1/2}$$

Pour imprimer des coefficients binomiaux (à l'américaine) ou d'autres structures semblables, on peut utiliser la commande `\binom` de l'extension amsmath :

```
\begin{displaymath}
\binom{n}{k}\qquad \mathrm{C}_n^k
\end{displaymath}
```

$$\binom{n}{k} \qquad \mathrm{C}_n^k$$

Il est parfois utile de pouvoir superposer des symboles, la commande `\stackrel` place son premier argument en taille réduite au dessus du second :

```
\begin{displaymath}
X \stackrel{!}{=} 1
\end{displaymath}
```

$$X \stackrel{!}{=} 1$$

Les **intégrales** sont produites par la commande `\int`, les **sommes** par la commande `\sum`, les produits par la commande `\prod`. Les limites inférieures et supérieures sont indiquées avec `_` et `^` comme pour les indices et les exposants.

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\prod_\epsilon
\end{displaymath}
```

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \prod_{\epsilon}$$

Pour superposer des indices, l'extension amsmath propose la commande `\substack` et l'environnement `subarray` qui permet d'aligner les indices à gauche au lieu de les centrer.

```
\begin{displaymath}
\sum_{\substack{0<i<n \\ 1<j<m}}
   P(i,j) =
\sum_{\begin{subarray}{l} i\in I\\
         1<j<m
      \end{subarray}}     Q(i,j)
\end{displaymath}
```

$$\sum_{\substack{0<i<n \\ 1<j<m}} P(i,j) = \sum_{\begin{subarray}{l} i\in I\\ 1<j<m \end{subarray}} Q(i,j)$$

Pour les **crochets et les autres délimiteurs**, il existe toutes sortes de symboles en TEX (par exemple $[\langle \| \updownarrow$). Les parenthèses et les crochets sont obtenus avec les caractères correspondants, les accolades avec `\{`, et tous les autres délimiteurs sont obtenus par des commandes spéciales (par exemple `\updownarrow`). Pour une liste de tous les délimiteurs disponibles, reportez-vous au tableau 3.8, page 55.

```
\begin{displaymath}
{a,b,c}\neq\{a,b,c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

Si vous ajoutez la commande `\left` avant un délimiteur ouvrant ou `\right` avant un délimiteur fermant, TEX détermine automatiquement la taille appropriée pour ce caractère. Remarquez qu'il est nécessaire de fermer chaque délimiteur ouvrant (`\left`) avec un délimiteur fermant (`\right`). Si vous ne voulez pas de délimiteur fermant, utilisez le délimiteur invisible `\right.` !

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
    \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{ 1-x^{2} } \right) ^3$$

Dans certains cas, il est nécessaire d'indiquer la taille exacte des délimiteurs mathématiques à la main. Vous pouvez alors utiliser les commandes `\big`, `\Big`, `\bigg` et `\Bigg` comme préfixes des commandes qui impriment les délimiteurs [3].

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$$\Big((x+1)(x-1) \Big) ^{2}$$
$$\big(\Big(\bigg(\Bigg(\quad \big\}\Big\}\bigg\}\Bigg\} \quad \big\|\Big\|\bigg\|\Bigg\|$$

Pour saisir des **points de suspension** dans une formule, vous pouvez utiliser plusieurs commandes. `\ldots` imprime les points sur la base de la ligne, `\cdots` les imprime au milieu. En plus il y a les commandes `\vdots` pour les imprimer verticalement et `\ddots` pour les imprimer en diagonale. Vous trouverez un autre exemple dans la section 3.5.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_1, \ldots, x_n \qquad x_1 + \cdots + x_n$$

[3] Ces commandes ne fonctionnent pas correctement après une commande de changement de taille ou si les options `11pt` ou `12pt` sont utilisées. Les extensions exscale ou amsmath permettent d'avoir le comportement attendu.

3.4 Espacement en mode mathématique

Si l'espacement choisi par TEX dans une formule n'est pas satisfaisant, il peut être ajusté en insérant des commandes d'espacement. Les plus importantes sont : `\,` pour une espace fine (␣), `\␣` pour une espace de taille moyenne (␣ représente le caractère « espace »), `\quad` (␣␣) et `\qquad` (␣␣␣␣) pour des espaces plus larges. La largeur d'un `\quad` correspond à la largeur du caractère « M » dans la police courante. La commande `\!` produit une espace fine négative (␣).

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
  \, \ud x\, \ud y
\end{displaymath}
plutôt que
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

$$\iint_D g(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

plutôt que

$$\int\int_D g(x,y)\mathrm{d}x\mathrm{d}y$$

Remarquez que le 'd' de l'élément différentiel est traditionnellement imprimé en caractères romains par la commande `\ud`.

$\mathcal{AMS}$-LaTeX propose un ensemble de commandes pour ajuster finement l'espace entre les signes intégrale : `\iint`, `\iiint`, `\iiiint` et `\idotsint`. Avec l'extension `amsmath`, l'exemple précédent peut se coder de cette manière :

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \, \ud x \, \ud y
\end{displaymath}
```

$$\iint_D \mathrm{d}x\,\mathrm{d}y$$

Reportez-vous au document `testmath.tex` distribué avec $\mathcal{AMS}$-LaTeX ou au chapitre 9 de *LaTeX, Apprentissage, guide et référence* [4] pour plus de détails.

3.5 Alignements verticaux

Pour imprimer des **matrices**, utilisez l'environnement `array`. Il fonctionne de manière similaire à l'environnement `tabular`. La commande `\\` est utilisée pour séparer les lignes.

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

L'environnement `array` peut également être utilisé pour imprimer des expressions qui ont un délimiteur invisible obtenu par la commande `\right.` :

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{si $d>c$}\\
 b+x & \textrm{le matin}\\
 l & \textrm{la journée}
  \end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{si } d > c \\ b + x & \textrm{le matin} \\ l & \textrm{la journée} \end{array} \right.$$

L'environnement `array` permet, comme l'environnement `tabular`, d'insérer des lignes horizontales ou verticales :

```
\begin{displaymath}
\left(\begin{array}{c|c}
 1 & 2 \\
\hline
3 & 4
\end{array}\right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}\right)$$

Pour les formules qui prennent plusieurs lignes ou pour des systèmes d'équations , utilisez les environnements `eqnarray` et `eqnarray*` plutôt que `equation`. Avec `eqnarray` chaque ligne est numérotée, alors que la variante `eqnarray*` ne produit aucun numéro.

Les environnements `eqnarray` et `eqnarray*` se comportent comme un tableau à trois colonnes de la forme `{rcl}`, où la colonne centrale peut être utilisée pour le signe égal, ou tout autre opérateur relationnel de votre choix. La commande `\\` sépare les lignes.

```
\begin{eqnarray}
f(x) & = & \cos x     \\
f'(x) & = & -\sin x   \\
\int_{0}^{x} f(y)\,dy &
 = & \sin x
\end{eqnarray}
```

$$\begin{aligned} f(x) &= \cos x && (3.5) \\ f'(x) &= -\sin x && (3.6) \\ \int_0^x f(y)\,dy &= \sin x && (3.7) \end{aligned}$$

On peut considérer qu'il y a trop d'espace de part et d'autre de la colonne centrale, autour des signes égal. Il peut être réduit par la commande `\setlength\arraycolsep{2pt}` comme dans l'exemple suivant.

Les **équations longues** ne sont pas découpées automatiquement en morceaux harmonieux. L'auteur doit indiquer où les couper et comment indenter la suite. Les deux méthodes ci-dessous sont les plus courantes pour obtenir le résultat attendu.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!} %-{}
                    \nonumber\\
 & & {}-\frac{x^{7}}{7!}+{}\cdots
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \tag{3.8}$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!}  }
                    \nonumber\\
 & & {}+\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!}+{}\cdots
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \tag{3.9}$$

La commande `\nonumber` empêche LaTeX de produire un numéro pour cette équation.

Il est parfois difficile d'obtenir des équations alignées verticalement correctement avec cette méthode. L'extension `amsmath` offre d'autres possibilités plus puissantes : voir notamment les environnements `multline`, `split` et `align` décrits dans [11], [3] et [4].

3.6 Fantômes...

Les fantômes sont invisibles mais permettent des ajustements intéressants en LaTeX.

Il arrive que LaTeX en fasse un peu trop dans des alignements verticaux d'indices ou d'exposants. La commande `\phantom` permet de réserver de l'espace pour des caractères invisibles, ce qui peut être utile comme le montrent les exemples suivants :

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\textrm{C}
\qquad \textrm{à comparer à} \qquad
{}^{12}_{6}\textrm{C}
\end{displaymath}
```

$$ {}^{12}_{6}\mathrm{C} \qquad \text{à comparer à} \qquad {}^{12}_{6}\mathrm{C} $$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\qquad \textrm{à comparer à} \qquad
\Gamma_{ij}^{k}
\end{displaymath}
```

$$\Gamma_{ij}{}^{k} \qquad \text{à comparer à} \qquad \Gamma_{ij}^{k}$$

3.7 Taille des polices mathématiques

En mode mathématique TEX choisit la taille de la police en fonction du contexte. Les exposants, par exemple, sont imprimés avec une police plus petite.

Malgré tout, il peut être nécessaire d'indiquer à LATEX la taille exacte. En mode mathématique, la taille de la police est déterminée par les quatre commandes :

`\displaystyle` (123), `\textstyle` (123), `\scriptstyle` (123) et `\scriptscriptstyle` (123).

Changer de style modifie également la façon dont les limites sont affichées.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
 \frac{\displaystyle
   \sum_{i=1}^n(x_i-\overline x)
   (y_i-\overline y)}
  {\displaystyle\biggl[
 \sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\operatorname{corr}(X,Y) = \frac{\displaystyle \sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\displaystyle \biggl[\sum_{i=1}^{n}(x_i-\overline{x})^2 \sum_{i=1}^{n}(y_i-\overline{y})^2\biggr]^{1/2}}$$

Cet exemple est un cas où on a besoin de crochets plus petits que ceux qui seraient produits par les commandes standards `\left[ \right]`.

3.8 Insertion de texte en mode mathématique

La commande `\mathrm` permet d'insérer du texte normal (en caractères romains) dans une formule. La taille du texte est ajustée automatiquement, mais on est limité à un mot et les accents sont interdits. La commande `\textrm` qui fait passer en mode *texte* permet l'utilisation des lettres accentuées et respecte les espaces. L'extension amsmath fournit une commande `\text` également très pratique pour insérer du texte dans une équation.

```
\begin{eqnarray*}
2^{\textrm{ième}}
&&\textrm{pour tout }x\\
2^{\mathrm{nd}}
&&\mathrm{pour tout }x\\
2^{\text{ième}}
&&\text{pour tout }x
\end{eqnarray*}
```

$$
\begin{aligned}
2^{\textrm{ième}} &\textrm{pour tout }x\\
2^{\mathrm{nd}} &\mathrm{pourtout}x\\
2^{\text{ième}} &\text{pour tout }x
\end{aligned}
$$

3.9 Théorèmes, propositions, etc.

En rédigeant des documents mathématiques, on a besoin d'un moyen de présenter des lemmes, des définitions, des axiomes et d'autres structures similaires. LaTeX prend cela en charge avec la commande :

`\newtheorem{`*nom*`}[`*compteur*`]{`*texte*`}[`*section*`]`

L'argument *nom* est un mot-clef utilisé pour identifier le théorème. L'argument *texte* définit le nom réel du théorème tel qu'il sera imprimé.

Les arguments entre crochets sont optionnels. Ils servent à indiquer la numérotation à utiliser sur le théorème. Avec *compteur* vous indiquez le *nom* d'un théorème déjà déclaré. Le nouveau théorème sera alors numéroté dans la même séquence. Avec *section* vous indiquez dans quel niveau de sectionnement vous voulez numéroter votre théorème.

Après avoir exécuté `\newtheorem` dans le préambule de votre document, vous pouvez utiliser la commande suivante :

`\begin{`*nom*`}[`*texte*`]`
Ceci est mon premier théorème
`\end{`*nom*`}`

Voilà pour la théorie. Les exemples qui suivent devraient montrer clairement que l'environnement `\newtheorem` est facile à utiliser.

```
% définitions dans le préambule
\newtheorem{loi}{Loi}
\newtheorem{decret}[loi]%
{Décret}
%dans le document
\begin{loi} \label{chef}
Le chef a raison.
\end{loi}
\begin{decret}[Important]
Le chef a toujours raison.
\end{decret}
\begin{loi}
Si le chef a tort, voir
la loi~\ref{chef}.
\end{loi}
```

Loi 1 *Le chef a raison.*

Décret 2 (Important) *Le chef a toujours raison.*

Loi 3 *Si le chef a tort, voir la loi 1.*

Le théorème « decret » utilise le même compteur que le théorème « loi ». C'est pourquoi il a un numéro dans la même séquence que les autres « lois ». L'argument entre crochets permet de spécifier un titre ou quelque chose de ce genre pour le théorème.

```
\newtheorem{mur}{Murphy}[section]
\begin{mur} Tout ce qui peut
aller mal ira mal.\end{mur}
```

Murphy 3.9.1 *Tout ce qui peut aller mal ira mal.*

Le théorème « Murphy » est numéroté à l'intérieur de la section en cours. On aurait pu utiliser un autre niveau tel que `chapter` ou `subsection`.

3.10 Symboles gras

Il est relativement difficile d'obtenir des symboles gras avec LaTeX ; cela est sans doute fait exprès car les typographes amateurs ont tendance à en abuser. La commande `\mathbf` permet d'obtenir des caractères gras, mais ce sont des caractères romains (droits), alors que les caractères mathématiques sont normalement en italique.

Les commandes `\mathversion{bold}` et et `\mathversion{normal}`, utilisables *en mode texte uniquement* permettent de changer le style par défaut utilisé par le mode mathématique.

```
\mathversion{bold}
\begin{displaymath}
\mu, M
\end{displaymath}
\mathversion{normal}
\begin{displaymath}
\mu, M \quad
\mathbf{\mu},\mathbf{M}
\end{displaymath}
```

$$\boldsymbol{\mu}, \boldsymbol{M}$$

$$\mu, M \quad \mu, \mathbf{M}$$

L'extension amsmath permet d'utiliser le gras pour un élément seulement dans une formule. Elle fournit la commande `\boldsymbol`.

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$$

3.11 Liste des symboles mathématiques

Dans les tableaux suivants, vous trouverez tous les symboles accessibles en mode *mathématique*.

Pour utiliser des symboles présents dans les tables 3.12 à 3.16 [4], l'extension amssymb doit être chargée dans le préambule du document et les polices mathématiques de l'AMS doivent être installées sur votre système. Si les extensions et les polices de l'AMS ne sont pas installées sur votre système, vous pouvez les récupérer sur
`CTAN:/macros/latex/packages/amslatex`

Tab. 3.1 – Accents en mode mathématique

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Tab. 3.2 – Alphabet grec minuscule

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

Tab. 3.3 – Alphabet grec majuscule

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

[4] Ces tables sont dérivées du fichier `symbols.tex` de David Carlisle et modifiées selon les suggestions de Josef Tkadlec

TAB. 3.4 – Relations binaires

Vous pouvez produire la négation de ces relations en préfixant ces commandes par `\not`.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` ou `\le`	$\geq$	`\geq` ou `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$ [a]	`\sqsubset` [a]	$\sqsupset$ [a]	`\sqsupset` [a]	$\bowtie$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` ou `\ne`

TAB. 3.5 – Opérateurs binaires

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$*$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lou`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$ [a]	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$ [a]	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

[a] Utilisez l'extension latexsym pour avoir accès à ces symboles

Tab. 3.6 – Opérateurs n-aires

$\sum$	\sum	$\bigcup$	\bigcup	$\bigvee$	\bigvee	$\bigoplus$	\bigoplus
$\prod$	\prod	$\bigcap$	\bigcap	$\bigwedge$	\bigwedge	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup			$\bigodot$	\bigodot
$\int$	\int	$\oint$	\oint			$\biguplus$	\biguplus

Tab. 3.7 – Flèches

$\leftarrow$	\leftarrow ou \gets	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\rightarrow$	\rightarrow ou \to	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\iff$	\iff (plus d'espace)	$\leadsto$ [a]	\leadsto [a]

[a] Utilisez l'extension latexsym pour obtenir ces symboles

Tab. 3.8 – Délimiteurs

(	(	)	)	$\uparrow$	\uparrow	$\Uparrow$	\Uparrow
[	[ou \lbrack	]	] ou \rbrack	$\downarrow$	\downarrow	$\Downarrow$	\Downarrow
{	\{ ou \lbrace	}	\} ou \rbrace	$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\langle$	\langle	$\rangle$	\rangle	$\vert$	| ou \vert	$\Vert$	\\| ou \Vert
$\lfloor$	\lfloor	$\rfloor$	\rfloor	$\lceil$	\lceil	$\rceil$	\rceil
/	/	\	\backslash				

Tab. 3.9 – Grands délimiteurs

$\lgroup$	\lgroup	$\rgroup$	\rgroup	$\lmoustache$	\lmoustache	$\rmoustache$	\rmoustache
$\arrowvert$	\arrowvert	$\Arrowvert$	\Arrowvert	$\bracevert$	\bracevert		

TAB. 3.10 – Symboles divers

$\ldots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$ [a]	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$ [a]	\Box [a]	$\Diamond$ [a]	\Diamond [a]
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg ou \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a] Utilisez l'extension latexsym pour obtenir ces symboles

TAB. 3.11 – Symboles non-mathématiques

Ces symboles peuvent également être utilisés en mode *texte*.

†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds

TAB. 3.12 – Délimiteurs de l'AMS

⌜	\ulcorner	⌝	\urcorner	⌞	\llcorner	⌟	\lrcorner

TAB. 3.13 – Caractères grecs et hébreux de l'AMS

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\daleth$	\daleth	$\gimel$	\gimel

TAB. 3.14 – Relations binaires de l'AMS

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot ou \Doteq
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll ou \llless	$\ggg$	\ggg ou \gggtr	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\Subset$	\Subset	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\therefore$	\therefore	$\because$	\because	$\varpropto$	\varpropto
$\shortmid$	\shortmid	$\shortparallel$	\shortparallel	$\between$	\between
$\smallsmile$	\smallsmile	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright	$\blacktriangleleft$	\blacktriangleleft
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq	$\blacktriangleright$	\blacktriangleright

TAB. 3.15 – Flèches de l'AMS

$\dashleftarrow$	\dashleftarrow	$\dashrightarrow$	\dashrightarrow	$\multimap$	\multimap
$\leftleftarrows$	\leftleftarrows	$\rightrightarrows$	\rightrightarrows	$\upuparrows$	\upuparrows
$\leftrightarrows$	\leftrightarrows	$\rightleftarrows$	\rightleftarrows	$\downdownarrows$	\downdownarrows
$\Lleftarrow$	\Lleftarrow	$\Rrightarrow$	\Rrightarrow	$\upharpoonleft$	\upharpoonleft
$\twoheadleftarrow$	\twoheadleftarrow	$\twoheadrightarrow$	\twoheadrightarrow	$\upharpoonright$	\upharpoonright
$\leftarrowtail$	\leftarrowtail	$\rightarrowtail$	\rightarrowtail	$\downharpoonleft$	\downharpoonleft
$\leftrightharpoons$	\leftrightharpoons	$\rightleftharpoons$	\rightleftharpoons	$\downharpoonright$	\downharpoonright
$\Lsh$	\Lsh	$\Rsh$	\Rsh	$\rightsquigarrow$	\rightsquigarrow
$\looparrowleft$	\looparrowleft	$\looparrowright$	\looparrowright	$\leftrightsquigarrow$	\leftrightsquigarrow
$\curvearrowleft$	\curvearrowleft	$\curvearrowright$	\curvearrowright		
$\circlearrowleft$	\circlearrowleft	$\circlearrowright$	\circlearrowright		

TAB. 3.16 – Négations des relations binaires et des flèches de l'AMS

$\not<$	\nless	$\not>$	\ngtr	$\not\subsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\not\supsetneqq$	\varsupsetneqq
$\not\leq$	\nleq	$\not\geq$	\ngeq	$\not\subseteqq$	\nsubseteqq
$\not\leqslant$	\nleqslant	$\not\geqslant$	\ngeqslant	$\not\supseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\not\leqq$	\nleqq	$\not\geqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\not\sim$	\nsim
$\not\prec$	\nprec	$\not\succ$	\nsucc	$\not\cong$	\ncong
$\not\preceq$	\npreceq	$\not\succeq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\not\subseteq$	\nsubseteq	$\not\supseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

TAB. 3.17 – Opérateurs binaires de l'AMS

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\Cup ou \doublecup	$\Cap$	\Cap ou \doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

TAB. 3.18 – Symboles divers de l'AMS

$\hbar$	`\hbar`	$\hslash$	`\hslash`	$\Bbbk$	`\Bbbk`
$\square$	`\square`	$\blacksquare$	`\blacksquare`	Ⓢ	`\circledS`
$\vartriangle$	`\vartriangle`	$\blacktriangle$	`\blacktriangle`	$\complement$	`\complement`
$\triangledown$	`\triangledown`	$\blacktriangledown$	`\blacktriangledown`	$\Game$	`\Game`
$\lozenge$	`\lozenge`	$\blacklozenge$	`\blacklozenge`	$\bigstar$	`\bigstar`
$\angle$	`\angle`	$\measuredangle$	`\measuredangle`	$\sphericalangle$	`\sphericalangle`
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`	$\backprime$	`\backprime`
$\nexists$	`\nexists`	$\Finv$	`\Finv`	$\varnothing$	`\varnothing`
$\eth$	`\eth`	$\mho$	`\mho`		

TAB. 3.19 – Polices mathématiques

Exemple	Commande	Extension à utiliser
ABCdef	`\mathrm{ABCdef}`	
$ABCdef$	`\mathit{ABCdef}`	
$ABCdef$	`\mathnormal{ABCdef}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	
$\mathscr{ABC}$	`\mathcal{ABC}`	mathrsfs
$\mathcal{ABC}$	`\mathcal{ABC}`	eucal avec l'option : mathcal ou
	`\mathscr{ABC}`	eucal avec l'option : mathscr
$\mathfrak{ABCdef}$	`\mathfrak{ABCdef}`	eufrak
$\mathbb{ABC}$	`\mathbb{ABC}`	amsfonts ou amssymb

Chapitre 4

Compléments

Pour rédiger un document important, LaTeX vous fournit des outils pour réaliser un index, une liste de références bibliographiques et d'autres choses. Des descriptions bien plus complètes de ces possibilités et des améliorations possibles avec LaTeX se trouvent dans le *LaTeX Manual* [1],dans *The LaTeX Companion* [3] et dans *LaTeX, Apprentissage, guide et référence* [4].

4.1 Figures PostScript

Avec les environnements `figure` et `table`, LaTeX fournit les mécanismes de base pour travailler avec des objets tels que des images ou des graphiques.

Il existe également plusieurs possibilités pour produire des graphiques avec des commandes LaTeX ou des extensions de LaTeX. Malheureusement, la plupart des utilisateurs trouvent ces commandes difficiles à mettre en œuvre. Pour plus d'informations à leur sujet, reportez-vous à *LaTeX, Apprentissage, guide et référence* [4] ou à *The LaTeX Companion* [3].

Un moyen bien plus simple d'inclure des graphiques dans un document est de les produire à l'aide d'un logiciel spécialisé[1] puis d'inclure le résultat dans le document. Ici encore LaTeX offre de nombreuses solutions. Dans cette introduction, seule l'utilisation de graphiques PostScript Encapsulé (EPS) sera traitée. En effet, c'est la solution la plus simple et la plus utilisée. Pour pouvoir utiliser des graphiques au format EPS, il faut disposer d'une imprimante acceptant les documents PostScript[2].

Un ensemble de commandes bien adaptées à l'insertion de graphiques est fourni par l'extension graphicx, développée par D. P. Carlisle. Elle fait partie d'un ensemble d'extensions appelé « graphics ». [3].

[1] Tel que XFig, CorelDraw!, FreeHand, Gnuplot...

[2] Une autre possibilité pour imprimer du PostScript est d'utiliser le logiciel GhostScript, disponible sur `CTAN:/support/ghostscript`

[3] `CTAN:/macros/latex/packages/graphics`

En supposant maintenant que vous travaillez sur un système où l'extension graphicx est installée et qui dispose d'une imprimante PostScript, voici la marche à suivre pour inclure une figure dans un document :

1. exportez la figure de votre logiciel graphique au format EPS ;
2. chargez l'extension graphicx dans le préambule de votre fichier source avec :

   `\usepackage[`*pilote*`]{graphicx}`

   pilote est le nom du programme de conversion « DVI vers PostScript » que vous utilisez. Le programme le plus répandu est `dvips`. Cette information est nécessaire car il n'y a pas de standard pour l'insertion de figures en TEX ; celle-ci est réalisée par le pilote au moment de l'impression. Connaître le nom du pilote permet à l'extension graphics d'inclure les bonnes commandes dans le fichier `.dvi` pour réaliser cette insertion ;
3. utilisez la commande :

   `\includegraphics[`*clef*`=`*valeur*`, ... ]{`*fichier*`}`

   pour insérer *fichier* dans votre document. Le paramètre optionnel est une liste de paires de *clefs* et de *valeurs* séparées par des virgules. Les *clefs* permettent de modifier la largeur, la hauteur, ou l'angle de rotation de la figure. Le tableau 4.1 présente les clefs les plus importantes.

TAB. 4.1 – Clefs pour l'extension graphicx

`width`	définit la largeur de la figure
`height`	définit la hauteur de la figure
`angle`	(en degrés) tourne la figure dans le sens des aiguilles d'une montre
`scale`	échelle de la figure

L'exemple suivant devrait illustrer le fonctionnement de la commande :

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=10cm]{test.eps}
\end{center}
\end{figure}
```

Cette commande inclut la figure stockée dans le fichier `test.eps`. La figure est *d'abord* tournée de 90 degrés puis ajustée pour que sa largeur finale

soit de 10 cm. Les proportions largeur/hauteur sont conservées, puisqu'aucune hauteur n'est spécifiée.

Pour plus d'informations, reportez vous à [9] et [14].

4.2 Références bibliographiques

L'environnement `thebibliography` permet de produire une liste de références bibliographiques. Chaque référence commence par

`\bibitem{`*marque*`}`

La *marque* est utilisée pour citer la référence dans le document.

`\cite{`*marque*`}`

La numérotation des références est produite automatiquement. Le paramètre qui suit `\begin{\thebibliography}` définit la largeur du décrochement utilisé pour placer ces numéros.

```
Partl~\cite{pa}
propose que\dots

{\small
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Vol.~9, No.~1 ('88)
\end{thebibliography}
}
```

Partl [1] propose que. . .

Bibliographie

[1] H. Partl: *German TEX*, TUGboat Vol. 9, No. 1 ('88)

Pour des projets plus importants, il est recommandé d'utiliser l'outil BiBTEX. Celui-ci est fourni avec la plupart des installations de TEX. Il permet de maintenir une base de données de références bibliographiques et d'en extraire la liste des références citées dans votre document. La génération des listes de références par BiBTEX utilise un mécanisme de feuilles de style qui permettent de réaliser tous les types de présentations habituellement demandés.

4.3 Index

L'index est un élément fort utile de nombreux ouvrages. LaTeX et le programme associé `makeindex`[4] permettent de créer des index assez facilement. Dans cette introduction, seules les commandes élémentaires de gestion d'un index sont présentées. Pour une description plus détaillée, reportez-vous à *The LaTeX Companion* [3]ou à *LaTeX, Apprentissage, guide et référence* [4].

Pour utiliser cette fonctionnalité, l'extension makeidx doit être chargée dans le préambule avec :

```
\usepackage{makeidx}
```

La création de l'index doit être activée par la commande :

```
\makeindex
```

placée dans le préambule.

Le contenu de l'index est défini par une série de commandes :

```
\index{clef}
```

où *clef* est un mot-clef de l'index. Vous insérez des commandes `\index` aux endroits du texte que vous voulez voir référencés par l'index. Le tableau 4.2 explique la syntaxe de l'argument *clef* avec plusieurs exemples.

Tab. 4.2 – Exemples de clefs d'index

Exemple	Résultat	Commentaires
`\index{hello}`	hello, 1	Entrée normale
`\index{hello!Peter}`	Peter, 3	Sous-entrée de 'hello'
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	Entrée formatée
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	idem
`\index{Jenny|textbf}`	Jenny, **3**	Numéro de page formaté
`\index{Joe|textit}`	Joe, *5*	idem
`\index{eolienne@éolienne}`	éolienne, 4	clef accentuée

Quand le fichier source est traité par LaTeX, chaque commande `\index` crée une entrée adaptée contenant le numéro de la page en cours dans le fichier qui porte le même nom de base que le fichier source, mais avec le

[4] sur les systèmes qui ne supportent pas les noms de fichiers de plus de huit caractères, ce programme s'appelle `makeidx`.

suffixe `.idx`. Ce fichier est ensuite traité par le programme `makeindex`.

`makeindex` *nom de fichier*

Le programme `makeindex` crée un index trié dans le fichier `.ind`. Ensuite, la prochaine fois que le fichier source sera traité, le contenu du fichier `.ind` sera inclus à l'endroit où LaTeX rencontrera la commande :

`\printindex`

L'extension showidx permet de visualiser les entrées de l'index dans la marge gauche du texte. Cela permet la relecture et la mise au point de l'index.

Le programme `makeindex` standard ne traite malheureusement pas correctement les caractères accentués dans les clefs : il les place systématiquement en tête de l'ordre alphabétique. Pour obtenir un classement correct des clés contenant des caractères accentués (le «é» doit être classé comme un «e»), on peut utiliser le caractère `@` : la dernière ligne du tableau 4.2 produira une entrée « éolienne » dans l'index, classée comme s'il s'agissait de « eolienne ».

4.4 En-têtes améliorés

L'extension fancyhdr[5], développée par P. van Oostrum, offre quelques commandes simples permettant de personnaliser les en-têtes et les pieds de page de votre document. Si vous regardez en haut de cette page, vous verrez un résultat possible de cette extension.

La difficulté principale pour personnaliser les en-têtes et les pieds de page consiste à mettre à jour le nom de la section ou du chapitre en cours utilisés par ces éléments. LaTeX réalise cela en deux étapes. Dans la définition des en-têtes et pieds de page les commandes `\leftmark` et `\rightmark` sont utilisées pour désigner respectivement les noms du chapitre et la section courants. La valeur de ces commandes est redéfinie chaque fois qu'un nouveau chapitre ou une nouvelle section commence.

Pour plus de souplesse, la commande `\chapter` et ses collègues ne redéfinissent pas `\leftmark` et `\rightmark` directement. Elles appellent les commandes appelées `\chaptermark`, `\sectionmark` et `\subsectionmark` qui sont chargées de redéfinir `\leftmark` et/ou `\rightmark`, selon la présentation désirée.

Ainsi, si vous voulez modifier la présentation du nom du chapitre courant dans l'en-tête, vous devrez redéfinir la commande `\chaptermark`.

[5] disponible sur `CTAN:/macros/latex/contrib/supported/fancyhdr/`.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% Ceci permet d'avoir les noms de chapitre et de section
% en minuscules
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}     % supprime les en-têtes et pieds prédéfinis
\fancyhead[LE,RO]{\bfseries\thepage}% Left Even, Right Odd
\fancyhead[LO]{\bfseries\rightmark} % Left Odd
\fancyhead[RE]{\bfseries\leftmark}  % Right Even
\renewcommand{\headrulewidth}{0.5pt}% filet en haut de page
\addtolength{\headheight}{0.5pt}    % espace pour le filet
\renewcommand{\footrulewidth}{0pt}  % pas de filet en bas
\fancypagestyle{plain}{ % pages de tetes de chapitre
   \fancyhead{}         % supprime l'entete
   \renewcommand{\headrulewidth}{0pt} % et le filet
}
```

FIG. 4.1 – Exemple de configuration de l'extension fancyhdr

La figure 4.1 montre un exemple de configuration de l'extension fancyhdr qui se rapproche de la présentation utilisée pour ce document. La documentation complète de cette extension se trouve à l'adresse mentionnée dans la note de bas de page.

4.5 L'extension verbatim

Plus haut dans ce document, vous avez appris à utiliser l'*environnement* verbatim. Dans cette section vous allez découvrir l'*extension* verbatim. L'extension verbatim est une nouvelle implémentation de l'environnement du même nom qui corrige certaines de ses limitations. En soi cela n'est pas spectaculaire, mais en plus, une fonctionnalité nouvelle a été ajoutée, et c'est elle qui justifie que cette extension soit citée ici. L'extension verbatim propose la commande :

`\verbatiminput{`*nom de fichier*`}`

qui permet d'inclure un fichier ASCII brut dans votre document, comme s'il se trouvait à l'intérieur d'un environnement verbatim.

Puisque l'extension verbatim fait partie de l'ensemble « tools », elle devrait être installée sur la plupart des systèmes. Pour en savoir plus au sujet de cette extension, reportez-vous à [10].

4.6 Téléchargement et installation d'extensions

La plupart des installations LaTeX fournissent en standard un grand nombre d'extensions, mais il arrive que justement celle dont on aurait besoin manque, ou qu'une extension nécessite une mise à jour. L'endroit le plus adéquat pour rechercher les versions officielles des extensions LaTeX est http://www.tex.ac.uk/CTANfind.html

Les extensions sont en général fournies sous la forme de deux fichiers, l'un de suffixe `.dtx`, l'autre de suffixe `.ins`. Souvent un fichier `readme.txt` est joint et donne une brève description de l'extension, le mieux est de commencer par la lecture de ce fichier.

Ensuite, il faut, d'une part installer l'extension et d'autre part extraire la documentation, voici la manière de procéder :

1. exécuter LaTeX sur le fichier `.ins`. Ceci produira les fichiers `.sty`, `.def`, etc., dont LaTeX a besoin.
2. déplacer ces fichiers dans un répertoire adéquat, en général c'est dans `.../texmf/tex/latex` ou dans `.../`*`localtexmf`*`/tex/latex`.
3. mettre à jour la base de données des noms de fichiers, la commande dépend de votre distribution LaTeX, c'est en général `mktexlsr texhash`, ou encore `texconfig rehash`.

Il faut ensuite extraire la documentation du fichier `.dtx` :

1. exécuter LaTeX sur le fichier `.dtx`. Cela produira un fichier `.dvi`. Noter que plusieurs exécutions de LaTeX peuvent être nécessaires pour produire les références croisées complètes.
2. vérifier si LaTeX a produit un fichier `.idx`. Si ce n'est pas le cas, passer à l'étape 5.
3. pour produire l'index, exécuter la commande suivante :

   ```
   makeindex -s gind.ist nom
   ```

   (où *nom* représente le nom du fichier principal, sans suffixe).
4. exécuter LaTeX sur le fichier `.dtx` une fois de plus.
5. enfin, produire un fichier PostScript ou PDF à imprimer pour une lecture plus confortable.

Parfois vous constaterez qu'un fichier `.glo` (glossaire) a également été produit. Exécutez la commande suivante entre les étapes 4 et 5 :

`makeindex -s gglo.ist -o` *nom*`.gls` *nom*`.glo`

Et n'oubliez pas de re-exécuter LaTeX sur le fichier `.dtx` avant de passer à l'étape 5.

Chapitre 5

Personnalisation de LaTeX

Les documents produits avec les commandes que vous avez apprises jusqu'ici sont de très bonne qualité aux yeux d'un vaste public. Même s'ils manquent de fantaisie, ils obéissent à toutes les règles de l'art de la typographie, ce qui les rend agréables à lire.

Mais il y a des situations où LaTeX ne propose pas de commande ou d'environnement adapté à vos besoins, ou bien où le résultat obtenu par une commande existante peut ne pas répondre à votre attente.

Dans ce chapitre, vous allez avoir un aperçu sur la manière d'enrichir les commandes de LaTeX et de modifier la présentation par défaut.

5.1 Vos propres commandes, environnements et extensions

Vous avez sûrement constaté que toutes les commandes décrites dans cet ouvrage sont présentées dans un cadre et sont référencées dans l'index qui se trouve à la fin. Au lieu d'utiliser à chaque fois l'ensemble des commandes LaTeX nécessaires, nous avons créé une extension dans laquelle nous avons défini de nouvelles commandes et de nouveaux environnement adaptés à cet usage. Ainsi nous pouvons simplement écrire :

```
\begin{lscommand}
\ci{dum}
\end{lscommand}
```

Dans cet exemple, nous utilisons à la fois un nouvel environnement appelé `lscommand` qui est responsable du tracé du cadre et une nouvelle commande appelée `\ci` qui compose le nom de la commande et qui produit l'entrée correspondante dans l'index. Vous pouvez le vérifier en cherchant la commande `\dum` dans l'index à la fin de ce document ; vous y trouverez une entrée pointant vers chaque page où la commande dum est mentionnée.

Si nous décidons un jour que nous ne souhaitons plus voir de cadre autour du nom des commandes, il nous suffira de modifier la définition de l'environnement `lscommand` pour définir un nouveau style. C'est bien plus simple (et efficace) que de parcourir tout le document pour remplacer une à une toutes les commandes qui tracent les cadres.

5.1.1 Nouvelles commandes

Pour ajouter de nouvelles commandes, utilisez la commande :

`\newcommand{`*nom*`}[`*num*`]{`*définition*`}`

Cette commande prend principalement deux arguments : le *nom* de la commande à créer et sa *définition*. L'argument *num* entre crochets est optionnel. Il indique le nombre de paramètres qu'utilisera la nouvelle commande (jusqu'à 9).

Les deux exemples ci-dessous vous aiderons à saisir le principe. Le premier exemple définit une nouvelle commande appelée `\ucil` qui est une abréviation de « une courte introduction à LaTeX 2ε ». Une telle commande pourrait être utile si vous aviez à citer de nombreuses fois le titre de ce livre.

```
\newcommand{\ucil}
    {Une courte (?)
     introduction à \LaTeXe}
% dans le document :
Voici \og \ucil \fg\dots
```

Voici « Une courte (?) introduction à LaTeX 2ε »...

L'exemple suivant montre comment utiliser l'argument *num*. La séquence `#1` est remplacée par le paramètre réel. Pour utiliser plus d'un paramètre, continuez avec `#2`, etc.

```
\newcommand{\uxil}[1]
    {Une \emph{#1}
     introduction à \LaTeXe}
% dans le document :
\begin{itemize}
\item \uxil{courte}
\item \uxil{rapide}
\end{itemize}
```

- Une *courte* introduction à LaTeX 2ε
- Une *rapide* introduction à LaTeX 2ε

LaTeX ne vous permet pas de créer une nouvelle commande si celle-ci existe déjà. Si vous voulez explicitement remplacer une commande existante, utilisez `\renewcommand`. Elle utilise la même syntaxe que `\newcommand`.

Dans certains cas, vous pouvez avoir besoin de `\providecommand`. Cette commande fonctionne comme `\newcommand`, mais si la nouvelle commande est déjà définie, LaTeX 2ε ignore la nouvelle définition.

LaTeX supprime en général les espaces qui suivent une commande (voir 6).

5.1.2 Nouveaux environnements

De manière analogue à la commande `\newcommand`, il est possible de définir de nouveaux environnements. La commande `\newenvironment` se présente de la manière suivante :

`\newenvironment{`*nom*`}[`*num*`]{`*avant*`}{`*après*`}`

De même que pour `\newcommand`, vous pouvez utiliser `\newenvironment` avec un paramètre optionnel ou sans. Le contenu de l'argument *avant* est exécuté avant que le contenu de l'environnement ne soit traité. Le contenu de l'argument *après* est traité lorsque l'on rencontre la commande `\end{`*nom*`}`.

L'exemple ci-dessous illustre l'utilisation de `\newenvironment`.

```
\newenvironment{king}
 {\rule{1ex}{1ex}%
      \hspace{\stretch{1}}}
 {\hspace{\stretch{1}}%
      \rule{1ex}{1ex}}

\begin{king}
Mes chers sujets, \dots
\end{king}
```

■ Mes chers sujets, . . . ■

L'argument *num* est utilisé de la même façon que pour la commande `\newcommand`. LaTeX vous empêche de redéfinir un environnement qui existe déjà. Si jamais vous vouliez redéfinir un environnement existant, utilisez `\renewenvironment` qui utilise la même syntaxe que `\newenvironment`.

Les commandes utilisées dans l'exemple ci-dessus seront présentées plus loin : pour la commande `\rule`, voir page 83, pour `\stretch`, voir page 77 enfin, pour plus d'informations sur `\hspace`, voir page 77.

5.1.3 Votre propre extension

Si vous définissez plein de nouveaux environnements et de nouvelles commandes, le préambule de votre document va se rallonger dangereusement. Il peut alors devenir intéressant de créer une extension contenant toutes ces nouvelles définitions. Avec la commande `\usepackage` vous pourrez rendre disponible votre extension dans votre document.

Écrire une extension consiste principalement à copier le préambule de votre document dans un fichier à part, dont le nom se termine par `.sty`. Il y a une commande spécifique à utiliser sur la première ligne de votre extension :

`\ProvidesPackage{`*nom de l'extension*`}`

`\ProvidesPackage` indique à LaTeX 2_ε le nom de l'extension afin notamment

```
% Exemple d'extension par Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\ucil}{Une courte (?) introduction à \LaTeXe}
\newcommand{\uxil}[1]{Une \emph{#1}
                      introduction à \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

FIG. 5.1 – Exemple d'extension

de lui permettre de produire des messages d'erreur significatifs. La figure 5.1 montre un exemple d'extension simple qui reprend les commandes définies dans les exemples précédents.

5.2 Polices et tailles des caractères

5.2.1 Commandes de changement de police

LaTeX choisit la police de caractères et sa taille en fonction de la structure logique du document (sections, notes de bas de page, etc.). Dans certains cas, on voudrait pouvoir changer la taille de la police à la main. Pour cela, utilisez les commandes listées dans les tableaux 5.1 et 5.2. La taille exacte de chaque police est un choix qui dépend de la classe de document et de ses options. La table 5.3 donne les tailles absolues en points pour les commandes présentes dans les classes de document standard.

```
{\small Les romains
petits et \textbf{gras}
régnèrent sur}
{\Large toute la belle
et grande \textit{Italie}.}
```

Les romains petits et **gras** régnèrent sur
toute la belle et grande *Italie*.

Une caractéristique importante de LaTeX 2_ε est que les différents attributs d'une police sont indépendants. Cela signifie que vous pouvez exécuter des commandes de changement de taille ou même de changement de police tout en conservant l'attribut gras ou italique.

En mode *mathématique*, vous pouvez utiliser les commandes de changement de police pour quitter provisoirement le mode mathématique et saisir du texte normal. Pour changer les attributs de la police en mode mathématique, il existe un jeu de commandes spéciales. Reportez-vous au tableau 5.4.

En complément des commandes de changement de taille, les accolades jouent un rôle essentiel. Elles sont utilisées pour former des *groupes* qui limitent la portée de la plupart des commandes de LaTeX.

Tab. 5.1 – Polices

`\textrm{...}`	roman	`\textsf{...}`	sans serif
`\texttt{...}`	`typewriter`		
`\textmd{...}`	medium	`\textbf{...}`	**bold face**
`\textup{...}`	upright	`\textit{...}`	*italic*
`\textsl{...}`	*slanted*	`\textsc{...}`	SMALL CAPS
`\emph{...}`	*emphasised*	`\textnormal{...}`	document font

Tab. 5.2 – Tailles des polices

`\tiny`	minuscule	`\Large`	plus grand
`\scriptsize`	très petit	`\LARGE`	très grand
`\footnotesize`	assez petit	`\huge`	énorme
`\small`	petit	`\Huge`	géant
`\normalsize`	normal		
`\large`	grand		

Tab. 5.3 – Tailles en points dans les classes standard

taille	10pt (défaut)	option 11pt	option 12pt
`\tiny`	5pt	6pt	6pt
`\scriptsize`	7pt	8pt	8pt
`\footnotesize`	8pt	9pt	10pt
`\small`	9pt	10pt	11pt
`\normalsize`	10pt	11pt	12pt
`\large`	12pt	12pt	14pt
`\Large`	14pt	14pt	17pt
`\LARGE`	17pt	17pt	20pt
`\huge`	20pt	20pt	25pt
`\Huge`	25pt	25pt	25pt

TAB. 5.4 – Polices mathématiques

Commande	*Exemple*	*Résultat*
`\mathcal{...}`	`$\mathcal{B}=c$`	$\mathcal{B} = c$
`\mathrm{...}`	`$\mathrm{K}_2$`	K_2
`\mathbf{...}`	`$\sum x=\mathbf{v}$`	$\sum x = \mathbf{v}$
`\mathsf{...}`	`$\mathsf{G\times R}$`	$\mathsf{G \times R}$
`\mathtt{...}`	`$\mathtt{L}(b,c)$`	$\mathtt{L}(b, c)$
`\mathnormal{...}`	`$\mathnormal{R_{19}}` `\neq R_{19}$`	$R_{19} \neq R_{19}$
`\mathit{...}`	`$\mathit{ffi}\neq ffi$`	$\mathit{ffi} \neq ffi$

```
Il aime les  {\LARGE grands et
{\small les petits}
caractères}.
```

Il aime les grands et les petits caractères.

Les commandes de changement de taille modifient également l'interligne, mais seulement si le paragraphe se termine dans la portée de la commande de changement de taille. L'accolade fermante `}` ne doit donc pas être placée trop tôt. Remarquez la position de la commande `\par` dans les deux exemples suivants [1] :

```
{\Large Ne lisez pas ceci ! Ce
n'est pas vrai ! Croyez-moi !\par}
```

Ne lisez pas ceci ! Ce n'est pas vrai ! Croyez-moi !

```
{\Large Ce n'est pas vrai. Mais
n'oubliez pas que je suis un
menteur.}\par
```

Ce n'est pas vrai. Mais n'oubliez pas que je suis un menteur.

Si vous voulez utiliser une commande de modification de la taille pour tout un paragraphe ou même plus, vous pouvez utiliser la syntaxe des environnements à la place des commandes.

```
\begin{Large}
Ceci n'est pas vrai
Mais, qu'est-ce qui l'est
de nos jours\dots
\end{Large}
```

Ceci n'est pas vrai Mais, qu'est-ce qui l'est de nos jours...

[1] `\par` est équivalent à une ligne vide

Cela vous évitera d'avoir à compter les accolades fermantes.

5.2.2 Attention danger

Il est dangereux d'utiliser de telles commandes de changement explicite de police tout au long de vos documents, en effet ces commandes vont à l'encontre de la philosophie de LaTeX qui est de séparer les aspects logiques et visuels d'un document. Cela signifie que si vous voulez utiliser en plusieurs endroits la même commande de changement de style afin de mettre en valeur un type particulier d'information, vous devriez utiliser `\newcommand` pour définir une nouvelle commande en insérant ainsi la commande de changement de style dans une enveloppe logique.

```
% dans le préambule ou dans
% une extension :
\newcommand{\danger}[1]
  {\textbf{#1}}
% dans le document :
Défense d'\danger{entrer}.
Cette pièce contient une
\danger{machine} d'origine
inconnue.
```

Défense d'**entrer**. Cette pièce contient une **machine** d'origine inconnue.

Cette approche sera précieuse si vous décidez plus tard d'utiliser une autre représentation typographique du danger que `\textbf`. Elle évitera d'avoir à rechercher et remplacer une à une toutes les occurrences de `\textbf` correspondant à la notion de danger.

5.2.3 Un conseil

Pour conclure cette promenade au pays des commandes de changement de police, voici un (mauvais) conseil :

N'oubliez pas! *Plus* vOus utilisez de polices **dans** un `document` *Plus* il DEVIENT *lisible et bien présenté*.

5.3 Espacement

5.3.1 Entre les lignes

Pour utiliser un interligne plus grand pour un document, vous pouvez utiliser la commande

`\linespread{`*facteur*`}`

dans le préambule de votre document. Utilisez `\linespread{1.3}` pour un

interligne « un et demi » et `\linespread{1.6}` pour un « double » interligne. L'interligne par défaut est 1.

5.3.2 Mise en page d'un paragraphe

Il y a deux paramètres qui jouent sur l'apparence d'un paragraphe. En insérant une définition telle que :

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

dans le préambule, vous supprimez le retrait des débuts de paragraphe (1re définition) et vous augmentez l'espace entre deux paragraphes (2e définition). Attention, la deuxième définition a également une influence sur la table des matières : ses lignes deviennent également plus espacées. Pour éviter cela, vous pouvez déplacer ces commandes du préambule vers le corps du document, après la commande `\tableofcontents` (ou bien ne pas les utiliser du tout, car la typographie professionnelle préfère utiliser l'indentation plutôt que l'espacement pour séparer les paragraphes).

Pour indenter un paragraphe qui ne l'est pas, utilisez la commande :

`\indent`

au début du paragraphe[2]. Bien sûr cela ne marche que si `\parindent` n'est pas nul.

Pour créer un paragraphe sans indentation, utilisez :

`\noindent`

en tête du paragraphe.

L'option francais de l'extension babel modifie ici aussi les règles par défaut de LaTeX pour s'adapter aux règles françaises.

Il est possible de commencer un paragraphe par une lettrine en utilisant l'extension lettrine[3] :

`\lettrine[`*options*`]{`*lettrine*`}{`*texte*`}`

La lettrine de la page iii s'obtient par la commande :

```
\lettrine{C}{e document}
```

[2] Pour indenter systématiquement le premier paragraphe après le titre d'une section, utilisez l'extension indentfirst de l'ensemble `tools`.

[3] `CTAN:/macros/latex/contrib/supported/lettrine/`

5.3.3 Espacement horizontal

LaTeX détermine l'espacement entre les mots et les phrases automatiquement. Pour ajouter de l'espacement horizontal, utilisez :

`\hspace{`*longueur*`}`

Si une telle espace doit être conservée, même lorsqu'elle tombe en début ou en fin de ligne, utilisez `\hspace*`. Dans le cas le plus simple, *longueur* est simplement un nombre suivi d'une unité. Les unités les plus importantes sont listées dans le tableau 5.5.

```
Ceci\hspace{1.5cm}est une espace
de 1.5 cm.
```

Ceci est une espace de 1.5 cm.

La commande :

`\stretch{`*n*`}`

produit une espace élastique. Elle s'étend jusqu'à ce que tout l'espace libre sur la ligne soit occupé. Si deux commandes `\hspace{\stretch{`*n*`}}` sont exécutées sur la même ligne, les deux espaces s'étendent en fonction du facteur d'élasticité *n*.

```
x\hspace{\stretch{1}}%
x\hspace{\stretch{3}}x
```

x x x

5.3.4 Espacement vertical

L'espacement vertical entre les paragraphes, sections, sous-sections... est déterminé automatiquement par LaTeX. En cas de besoin, de l'espace supplémentaire *entre deux paragraphes* peut être inséré avec la commande :

`\vspace{`*longueur*`}`

Cette commande doit normalement être utilisée entre deux lignes vides. Si l'espacement doit être conservé en haut ou en bas d'une page, utilisez la version étoilée de la commande `\vspace*`.

La commande `\stretch` en association avec `\pagebreak` permet d'imprimer du texte sur la dernière ligne d'une page ou de centrer verticalement du texte sur une page.

```
Du texte \dots

\vspace{\stretch{1}}
Ceci sera imprimé sur la dernière ligne.\pagebreak
```

TAB. 5.5 – Unités TeX

`mm`	millimètre	⊔
`cm`	centimètre = 10 mm	⊔‾‾‾‾⊔
`in`	pouce [a] = 25,4 mm	⊔‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾⊔
`pt`	point ≈ 1/72 pouce ≈ 0,35 mm	‖
`em`	largeur d'un "M" dans la police courante	⊔‾⊔
`ex`	hauteur d'un "x" dans la police courante	⊔

[a] *Inch* en anglais. *NdT.*

De l'espace supplémentaire entre deux lignes du *même* paragraphe ou à l'intérieur d'une table peut être obtenu par la commande :

`\\[`*longueur*`]`

Les commandes `\bigskip` et `\smallskip` permettent de créer des espacements verticaux prédéfinis sans se préoccuper des dimensions exactes.

5.4 Disposition d'une page

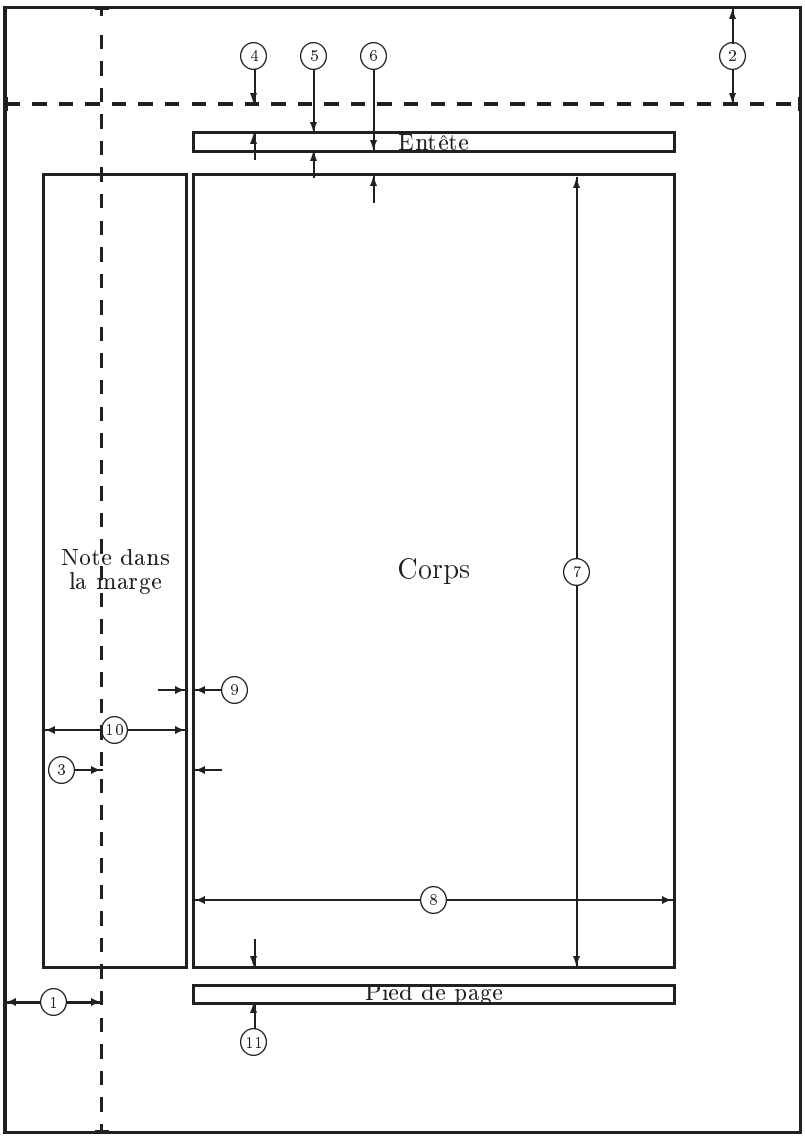
LaTeX 2_ε permet d'indiquer la taille du papier en paramètre de la commande `\documentclass`. Il définit ensuite automatiquement les marges les mieux adaptées. Parfois, on peut ne pas être satisfait par les valeurs prédéfinies et vouloir les modifier. La figure 5.2 montre tous les paramètres qui peuvent être modifiés. Cette figure a été réalisée avec l'extension `layout` de l'ensemble tools.

Attendez !... avant de vous lancer dans « élargissons un peu ce texte », prenez deux secondes pour réfléchir. Comme souvent avec LaTeX, il y a de bonnes raisons pour disposer les pages de cette façon.

Sans doute, comparé avec une page standard produite avec MS Word, une page de LaTeX à l'air horriblement étroite. Mais regardez votre livre préféré [4] et comptez le nombre de caractères sur une ligne normale. Vous verrez qu'il n'y a guère plus de soixante-six caractères par ligne. L'expérience montre qu'un texte devient moins lisible si le nombre de caractères par ligne dépasse cette valeur, cela parce qu'il devient plus difficile pour les yeux de passer de la fin d'une ligne au début de la ligne suivante. Ceci explique aussi que les journaux utilisent plusieurs colonnes.

Ainsi, si vous élargissez le corps du texte, ayez conscience que vous le rendez aussi moins lisible. Ceci dit, si vous tenez à modifier les paramètres qui contrôlent la disposition d'une page, voici comment procéder :

[4] Un vrai livre, imprimé par un grand éditeur...



1	un pouce + \hoffset	2	un pouce + \voffset
3	\evensidemargin = 70pt	4	\topmargin = 22pt
5	\headheight = 13pt	6	\headsep = 19pt
7	\textheight = 595pt	8	\textwidth = 360pt
9	\marginparsep = 7pt	10	\marginparwidth = 106pt
11	\footskip = 27pt		\marginparpush = 5pt (non affiché)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

FIG. 5.2 – Paramètres de la disposition d'une page

LaTeX dispose de deux commandes pour modifier ces paramètres. Elles sont généralement utilisées dans le préambule.

La première commande affecte une valeur fixe au paramètre spécifié :

`\setlength{`*paramètre*`}{`*longueur*`}`

La deuxième commande ajoute une longueur à ce paramètre.

`\addtolength{`*paramètre*`}{`*longueur*`}`

La deuxième commande est en pratique plus utile que `\setlength`, parce qu'elle permet de travailler en relatif par rapport à la taille par défaut. Pour ajouter un centimètre à la largeur du texte, nous utiliserions les commandes suivantes dans le préambule :

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

Dans ce contexte, il peut être intéressant d'utiliser l'extension `calc`, qui permet d'utiliser des expressions algébriques traditionnelles en argument de `setlength` ainsi que partout où l'on utilise des valeurs numériques comme arguments de fonctions.

5.5 Jouons un peu avec les dimensions

Autant que possible nous évitons d'utiliser des dimensions absolues dans des documents LaTeX. Nous essayons plutôt de les définir relativement aux dimensions d'autres éléments de la page. La largeur d'une figure sera ainsi `\textwidth` afin de lui faire occuper toute la largeur de la page.

Les trois commandes suivantes permettent de déterminer la largeur, la hauteur et la profondeur d'une chaîne de caractères.

`\settoheight{`*commande*`}{`*texte*`}`
`\settodepth{`*commande*`}{`*texte*`}`
`\settowidth{`*commande*`}{`*texte*`}`

L'exemple ci-dessous montre une utilisation possible de ces commandes :

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1\ }
  \makebox[0pt][r]{#1\ }}{}
\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Où :}$a$,
$b$ sont les cotés adjacents à
l'angle droit d'un triangle
rectangle,\par
$c$ est l'hypothénuse du
triangle,\par
$d$ n'est pas utilisé ici.
\'Etonnant non ?
\end{vardesc}
```

$$a^2 + b^2 = c^2$$

Où : a, b sont les cotés adjacents à l'angle droit d'un triangle rectangle,
c est l'hypothénuse du triangle,
d n'est pas utilisé ici. Étonnant non ?

5.6 Boîtes

LaTeX construit ses pages en empilant des boîtes. Au commencement chaque caractère est une petite boîte qui est ensuite collée à d'autres boîtes-caractères pour former un mot. Ceux-ci sont alors assemblés à d'autres mots, avec une colle spéciale qui est élastique pour permettre de comprimer ou d'étirer des séries de mots afin de remplir exactement une ligne sur la page.

Reconnaissons qu'il s'agit d'une description simpliste de ce qui se passe réellement, mais le fait est là : TeX travaille avec des boîtes et de la colle. Les caractères ne sont pas les seuls à pouvoir former des boîtes. Virtuellement tout peut être mis dans des boîtes, y compris d'autres boîtes. Chaque boîte est ensuite traitée par LaTeX comme s'il s'agissait d'un simple caractère.

Dans les chapitres précédents vous avez déjà rencontré quelques boîtes, même si nous ne l'avons pas signalé. L'environnement `tabular` et la commande `\includegraphics`, par exemple, produisent tous les deux des boîtes. Cela signifie que vous pouvez facilement aligner deux illustrations ou deux tables côte à côte. Il suffit de s'assurer que la somme de leurs largeurs ne dépasse pas la largeur du texte.

Il est aussi possible d'emballer un paragraphe dans une boîte :

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

on peut également utiliser un environnement :

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

le paramètre `pos` peut être l'une des lettres `c`, `t` ou `b` pour contrôler l'aligne-

ment vertical de la boîte par rapport à la base du texte précédent. `largeur` est une dimension indiquant la largeur de la boîte.

Alors que `\parbox` englobe tout un paragraphe en réalisant coupure des lignes et tout le reste, il existe également une catégorie de commandes de gestion des boîtes qui ne travaillent que sur des éléments alignés horizontalement. L'une d'elles nous est déjà connue. Il s'agit de `\mbox` qui combine simplement une série de boîtes pour en former une nouvelle; elle peut être utilisée pour empêcher LaTeX de couper une ligne entre deux mots. Puisqu'il est possible de placer des boîtes dans d'autres boîtes, ces constructeurs de boîtes horizontales sont extrêmement flexibles.

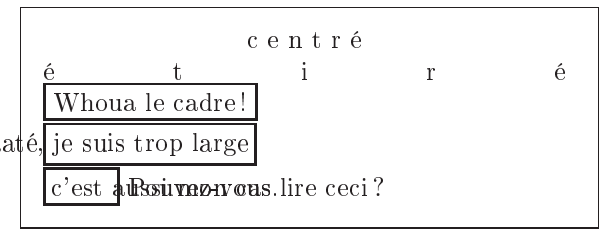
`\makebox[`*largeur*`][`*pos*`]{`*texte*`}`

Le paramètre `largeur` définit la largeur de la boîte vue de l'extérieur[5]. En plus des expressions exprimant une longueur vous pouvez également utiliser `\width`, `\height`, `\depth` et `\totalheight` à l'intérieur du paramètre `largeur`. Leurs valeurs sont obtenues à partir des dimensions réelles du `texte`. Le paramètre `pos` est une lettre parmi `c` (**c**enter) pour centrer le texte, `l` (**l**eft flush) pour l'aligner à gauche, `r` (**r**ight flush) pour l'aligner à droite, ou `s` (**s**pread) pour le répartir horizontalement dans la boîte.

La commande `\framebox` fonctionne de la même façon que `\makebox`, mais en plus elle dessine un cadre autour du texte.

L'exemple suivant vous montre quelques choses que l'on peut faire avec les commandes `\makebox` et `\framebox` :

```
\makebox[\textwidth]{%
    c e n t r \'e}\par
\makebox[\textwidth][s]{%
é t i r é}\par
\framebox[1.1\width]{Whoua
    le cadre !} \par
\framebox[0.8\width][r]{Rat\'e,
    je suis trop large} \par
\framebox[1cm][l]{c'est aussi
    mon cas.}
Pouvez-vous lire ceci ?
```



Maintenant que nous savons contrôler l'alignement horizontal, la suite logique est de voir comment gérer l'alignement vertical[6]. Pas de problème

[5] Cela signifie qu'elle peut être plus petite que la largeur du contenu de la boîte. Dans un cas extrême on peut même positionner la largeur à `0pt`; ainsi le texte dans la boîte sera placé sans influencer les boîtes adjacentes.

[6] Le contrôle total est obtenu en contrôlant en même temps l'alignement horizontal et l'alignement vertical.

avec LaTeX. La commande :

`\raisebox{`*élévation*`}[`*profondeur*`][`*hauteur*`]{`*texte*`}`

permet de définir les propriétés verticales d'une boîte. Vous pouvez utiliser `\width`, `\height`, `\depth` et `\totalheight` dans les trois premiers paramètres afin d'agir en fonction de la taille du texte contenu dans la boîte.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}}
cria-t-il, mais la ligne suivante
ne remarqua pas qu'une chose
horrible lui était arrivée.
```

Aaaa$_{\mathbf{aa}_{\mathbf{r}_{\mathbf{g}}}}$ cria-t-il, mais la ligne suivante ne remar**g**ua pas qu'une chose horrible lui était arrivée. **h**

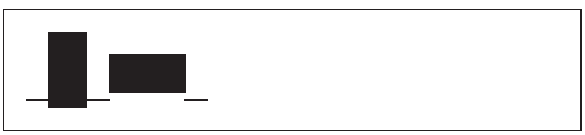
5.7 Filets

Quelques pages plus haut vous avez peut-être remarqué la commande :

`\rule[`*élévation*`]{`*largeur*`}{`*hauteur*`}`

En utilisation normale, elle produit une simple boîte noire.

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



C'est utile pour produire des lignes horizontales et verticales. La ligne horizontale sur la page de titre par exemple a été tracée à l'aide d'une commande `\rule`.

Un cas particulier est un filet avec une largeur nulle mais une certaine hauteur. En typographie professionnelle ceci est appelé un montant. Il sont utilisés pour s'assurer qu'un élément aura une hauteur donnée. Vous pouvez par exemple l'utiliser dans un environnement `tabular` pour garantir la hauteur d'une ligne.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}\'Etai\dots\\
\hline
\rule{0pt}{4ex} montant \\
\hline
\end{tabular}
```

Bibliographie

[1] Lamport, Leslie. *LaTeX: A Document Preparation System.* Addison-Wesley, 1994. 2e édition.
ISBN 0-201-52983-1.

[2] Knuth, Donald E. *The TeXbook,* Volume A de *Computers and Typesetting.* Addison-Wesley, 1984. 2e édition.
ISBN 0-201-13448-9.

[3] Goossens, Michel ; Mittelbach, Frank et Samarin, Alexander. *The LaTeX Companion.* Addison-Wesley, 1994.
ISBN 0-201-54199-8.

[4] Desgraupes, Bernard. *LaTeX Apprentissage, guide et référence.* Vuibert, 2000.
ISBN 2-7117-8658-7.

[5] Chaque installation de LaTeX devrait fournir un document appelé *LaTeX Local Guide* qui explique les particularités de cette installation. Malheureusement certains administrateurs système paresseux ne fournissent pas ce document. Dans ce cas, demandez de l'aide aux autres utilisateurs autour de vous ou au gourou local de LaTeX.

[6] LaTeX3 Project Team. *LaTeX 2ε for authors.* Distribué avec LaTeX 2ε dans `usrguide.tex`.

[7] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers.* Distribué avec LaTeX 2ε dans `clsguide.tex`.

[8] LaTeX3 Project Team. *LaTeX 2ε Font selection.* Distribué avec LaTeX 2ε dans `fntguide.tex`.

[9] Carlisle, David P. *Packages in the 'graphics' bundle.* Distribué avec les extensions « graphics » dans `grfguide.tex`.

[10] Schöpf, Rainer ; Raichle, Bernd et Rowley Chris. *A New Implementation of LaTeX's verbatim Environments.* Distribué avec l'ensemble « tools » dans `verbatim.dtx`.

[11] American Mathematical Society *AMS-LaTeX Version 1.2 User's guide.* Distribué avec les extensions AMS-LaTeX dans `amsldoc.tex`.

[12] Gaulle, Bernard. *Notice d'utilisation du style french multilingue.* Disponible avec l'extension `french` sur `http://frenchpro.free.fr/`.

[13] Perrousseaux, Yves. *Manuel de typographie française élémentaire.* Ateliers Perrousseaux éditeur, 1995.
ISBN 2-911220-00-5.

[14] Reckdahl, Keith. *Using EPS Graphics in LaTeX 2ε Documents* qui explique tout ce que vous avez toujours voulu savoir et même plus sur les fichiers PostScript et leur utilisation avec LaTeX. Disponible en ligne sur `CTAN:/info/epslatex.ps`

Index

Symboles

\! 46
" 19
$ 39
\(..... 39
\) 39
\, ... 40, 46
- 20
– 20
\- 18
– 20
— 20
., espace après 25
..... 21
.aux 13
.cls 13
.dtx 13
.dvi 13
.fd 13
.idx 14
.ilg 14
.ind 14
.ins 13
.lof 13
.log 13
.lot 13
.sty 13
.tex 13
.toc 13
\@ 25
\[..... 39
& 34
\\ 17, 31, 32, 34, 78
* 17
\] 39
~ 20, 25

A

A4 (papier) 10
A5 (papier) 10
accent 22
  aigu 22
  circonflexe 22
  grave 22
accolade 72
accolades 6, 44
\addtolength 80
æ 22
alignement décimal 34
allemand 25
amsfonts 41, 59
amsmath ... 39, 44–46, 48, 49, 52
amssymb 41, 53, 59
\and 27
antislash 6
\appendix 26, 28
array 46, 47
article (classe) 9
\author 27
avantages de LaTeX 4

B

B5 (papier) 10
babel 20–24, 26, 30, 76
\backmatter 28
backslash 6
\backslash 6
\begin 30
\bibitem 63
\Big 45
\big 45
\Bigg 45
\bigg 45

\bigskip 78
\binom 44
blackboard bold 41
blancs 5
\bmod 43
boîte 81
bold face 73
\boldsymbol 52
book (classe) 9

C

cédille 22
calc 80
\caption 37, 38
caractères réservés 5
caractères spéciaux 22
\cdot 43
\cdots 45
center 31
césure 18
\chapter 26
\chaptermark 65
\ci 69
\cite 63
\cleardoublepage 38
\clearpage 38
codage 12
commandes
\! 46
\(..... 39
\) 39
\, ... 40, 46
\- 18
\@ 25
\[..... 39
\\ 17, 31, 32, 34, 78
* 17
\] 39
\addtolength 80
\and 27
\appendix 26, 28
\author 27
\backmatter 28
\backslash 6
\begin 30
\bibitem 63
\Big 45
\big 45
\Bigg 45
\bigg 45
\bigskip 78
\binom 44
\bmod 43
\boldsymbol 52
\caption 37, 38
\cdot 43
\cdots 45
\chapter 26
\chaptermark 65
\ci 69
\cite 63
\cleardoublepage 38
\clearpage 38
\date 27
\ddots 45
\degres 21
\depth 82, 83
\displaystyle 49
\documentclass 9, 13
\dots 21
\dq 25
\dum 69
\emph 29, 73
\end 30
\fg 20
\footnote 28, 38
\footnotesize 73
\frac 43
\framebox 82
\frenchspacing 26
\frontmatter 27
\fussy 18
\height 82, 83
\hline 34
\hspace 71, 77
\Huge 73
\huge 73
\hyphenation 18

\idotsint 46
\iiiint 46
\iiint 46
\iint 46
\include 14
\includegraphics 62
\includeonly 14
\indent 76
\index 64
\input 14
\int 44
\item 30
\label 28, 40
\LARGE 73
\Large 73
\large 73
\LaTeX 19
\LaTeXe 19
\ldots 45
\left 45
\leftmark 65
\lettrine 76
\linebreak 17
\linespread 75
\listoffigures 37
\listoftables 37
\mainmatter 28
\makebox 82
\makeindex 64
\maketitle 27
\mathbb 41
\mathbf 74
\mathcal 74
\mathit 74
\mathnormal 74
\mathrm 49, 74
\mathsf 74
\mathtt 74
\mathversion 51
\mbox 19, 21, 82
\multicolumn 35
\newcommand 70, 75
\newenvironment 71
\newline 17
\newpage 17
\newtheorem 50
\noindent 76
\nolinebreak 17
\nonumber 48
\nopagebreak 17
\normalsize 73
\og 20
\overbrace 42
\overleftarrow 43
\overline 42
\overrightarrow 43
\pagebreak 17
\pageref 28
\pagestyle 11
\par 74
\paragraph 26
\parbox 81, 82
\parindent 76
\parskip 76
\part 26
\phantom 38, 48
\pmod 43
\printindex 65
\prod 44
\protect 38
\providecommand 70
\ProvidesPackage 71
\qquad 40, 46
\quad 40, 46
\raisebox 83
\ref 28, 40
\renewcommand 70
\renewenvironment 71
\right 45
\right. 45, 47
\rightmark 65
\rule 71, 83
\scriptscriptstyle 49
\scriptsize 73
\scriptstyle 49
\section 26, 38
\sectionmark 65
\setlength 76, 80

\settodepth 80
\settoheight 80
\settowidth 80
\sim 20
\sloppy 18
\small 73
\smallskip 78
\sqrt 42
\stackrel 44
\stretch 71, 77
\subparagraph 26
\subsection 26
\subsectionmark 65
\substack 44
\subsubsection 26
\sum 44
\tableofcontents ... 27, 76
\TeX 19
\text 49
\textbf 73
\textit 73
\textmd 73
\textnormal 73
\textrm 49, 73
\textsc 73
\textsf 73
\textsl 73
\textstyle 49
\texttt 73
\textup 73
\thispagestyle 11
\tiny 73
\title 27
\today 19
\totalheight 82, 83
\ud 46
\underbrace 42
\underline 29, 42
\url 20
\usepackage 11, 23, 71
\vdots 45
\vec 42
\verb 33
\verbatiminput 66
\vspace 77
\widehat 42
\widetilde 42
\width 82, 83
comment 7
commentaires 6
crochets 6, 44
CTAN vi

D

délimiteurs 44
\date 27
dcolumn 34
\ddots 45
degré 20
\degres 21
\depth 82, 83
description 30
deutsch 25
deux colonnes 10
dimensions 77
displaymath 39
\displaystyle 49
disposition d'une page 78
\documentclass 9, 13
\dots 21
double interligne 76
double-face 10
\dq 25
\dum 69
dvips 8

E

\emph 29, 73
emplacement 36
empty 11
en-tête 11
\end 30
enumerate 30
environnement 30
environnements
array 46, 47
center 31
comment 7

description 30
displaymath 39
enumerate 30
eqnarray 47
equation 40
figure 35–37, 61
flushleft 31
flushright 31
itemize 30
lscommand 69, 70
math 39
minipage 81
quotation 32
quote 31
subarray 44
table 36, 37
tabular 33
thebibliography 63
verbatim 32, 66
verse 32
eqnarray 47
equation 40
équations
longues 48
système d' 47
espace 5
après une commande 6
en début de ligne 5
espace insécable 25
espacement
horizontal 77
mathématique 46
vertical 77
eucal 59
eufrak 59
executive (papier) 10
exposant 42
exscale 12, 45
extension 7, 11
extensions
amsfonts 41, 59
amsmath .. 39, 44–46, 48, 49, 52
amssymb 41, 53, 59
babel 20–24, 26, 30, 76
calc 80
dcolumn 34
eucal 59
eufrak 59
exscale 12, 45
fancyhdr 65, 66
fontenc 12, 24
francais 20, 21, 24, 26, 30, 76
french 23
graphics 62
graphicx 61
hyperref 20
ifthen 12
indentfirst 76
inputenc 12, 23
latexsym 12
layout 78
lettrine 76
longtabular 35
makeidx 12, 64
mathrsfs 59
showidx 65
supertabular 35
syntonly 12
verbatim 7, 66

F

fancyhdr 65, 66
\fg 20
fichier source 5
figure 35
figure 35–37, 61
filet 83
fleqn 10
flushleft 31
flushright 31
foiltex 9
fontenc 12, 24
\footnote 28, 38
\footnotesize 73
formules 39
\frac 43
fraction 43

\framebox 82
francais 20, 21, 24, 26, 30, 76
french 23
\frenchspacing 26
\frontmatter 27
\fussy 18

G

graphics 62
graphicx 61
graphiques 61
grec
    alphabet 41
groupe 72
guillemets 19

H

headings 11
\height 82, 83
\hline 34
horizontal
    accolade 42
    filet 83
    trait 42
\hspace 71, 77
\Huge 73
\huge 73
hyperref 20
Hyphenation 18
\hyphenation 18

I

ı et ȷ sans points 22
\idotsint 46
ifthen 12
\iiiint 46
\iiint 46
\iint 46
\include 14
\includegraphics 62
\includeonly 14
\indent 76
indentfirst 76
index 64
\index 64
indice 42
\input 14
inputenc 12, 23
\int 44
intégrale 44
interligne 75
international 22
italic 73
\item 30
itemize 30

J

justification 17

K

Knuth, Donald E. 1

L

\label 28, 40
Lamport, Leslie 1
\LARGE 73
\Large 73
\large 73
\LaTeX 19
LaTeX 2.09 2
LaTeX 2ε 2
LaTeX3 ..... 2, 4
\LaTeXe 19
latexsym 12
layout 78
\ldots 45
\left 45
\leftmark 65
legal (papier) 10
leqno 10
letter (papier) 10
lettrine 76
\lettrine 76
ligature 21
\linebreak 17
\linespread 75
\listoffigures 37
\listoftables 37
livre 9
longtabular 35

lscommand 69, 70

M
\mainmatter 28
\makebox 82
makeidx 12, 64
\makeindex 64
makeindex 65
\maketitle 27
marges 78
math 39
mathématiques 39
accents 42
espaces 46
fonctions 43
\mathbb 41
\mathbf 74
\mathcal 74
\mathit 74
\mathnormal 74
\mathrm 49, 74
mathrsfs 59
\mathsf 74
\mathtt 74
\mathversion 51
\mbox 19, 21, 82
minipage 81
Mittelbach, Frank 2
modulo 43
moins (signe) 20
montant 83
\multicolumn 35

N
\newcommand 70, 75
\newenvironment 71
\newline 17
\newpage 17
\newtheorem 50
\noindent 76
\nolinebreak 17
\nonumber 48
\nopagebreak 17
\normalsize 73
notitlepage 10

O
objets flottants 35
œ 22
\og 20
oneside 10
openany 10
openright 10
option 9
\overbrace 42
overfull hbox 18
\overleftarrow 43
\overline 42
\overrightarrow 43

P
\pagebreak 17
\pageref 28
\pagestyle 11
papier
A4 10
A5 10
letter 10
taille du 10
\par 74
\paragraph 26
paragraphe 15
paramètre 6
paramètres optionnels 6
\parbox 81, 82
parenthèses 44
\parindent 76
\parskip 76
\part 26
\phantom 38, 48
pieds de page 11
plain 11
\pmod 43
point 21
points de suspension 21, 45
diagonaux 45
verticaux 45
police 72

PostScript Encapsulé 61
préambule 7
prime 42
\printindex 65
\prod 44
\protect 38
\providecommand 70
\ProvidesPackage 71

Q

\qquad 40, 46
\quad 40, 46
quotation 32
quote 31

R

références croisées 28
racine carrée 42
\raisebox 83
rapport 9
recto simple 10
recto-verso 10
\ref 28, 40
\renewcommand 70
\renewenvironment 71
report (classe) 9
\right 45
\right. 45, 47
\rightmark 65
roman 73
\rule 71, 83

S

sans serif 73
scandinaves (caractères) 22
\scriptscriptstyle 49
\scriptsize 73
\scriptstyle 49
\section 26, 38
\sectionmark 65
\setlength 76, 80
\settodepth 80
\settoheight 80
\settowidth 80
showidx 65
\sim 20
simple face 10
slanted 73
slides 9
\sloppy 18
\small 73
small caps 73
\smallskip 78
somme 44
\sqrt 42
\stackrel 44
\stretch 71, 77
style de page
    empty 11
    headings 11
    plain 11
subarray 44
\subparagraph 26
\subsection 26
\subsectionmark 65
\substack 44
\subsubsection 26
\sum 44
supertabular 35
symboles
    gras 51
    mathématiques 53
syntonly 12
systèmes d'équations 47

T

table 36, 37
tableau 35
\tableofcontents 27, 76
tabular 33
taille
    de la police par défaut ... 10
    des polices 72
    des polices mathématiques .. 49
    du papier 10
    prédéfinies 73
\TeX 19
\text 49

\textbf 73
\textit 73
\textmd 73
\textnormal 73
\textrm 49, 73
\textsc 73
\textsf 73
\textsl 73
\textstyle 49
\texttt 73
\textup 73
thebibliography 63
\thispagestyle 11
tilde 20, 25, 42
\tiny 73
tiret 20
\title 27
titlepage 10
titre du document 10, 27
\today 19
\totalheight 82, 83
transparents 9
twocolumn 10
twoside 10

U
\ud 46
umlaut 22
\underbrace 42
underfull hbox 18
\underline 29, 42
unités 77, 78
upright 73
URL 20
\url 20
\usepackage 11, 23, 71

V
\vdots 45
\vec 42
vecteurs 42
\verb 33
verbatim 7, 66
verbatim 32, 66
\verbatiminput 66
verse 32
vertical
espacement 77
points de suspension 45
virgule 21
\vspace 77

W
\widehat 42
\widetilde 42
\width 82, 83
www 20
WYSIWYG 3, 4

X
xdvi 8