Une courte (?) introduction à $\LaTeX\,2_\varepsilon$

ou LaTeX2e en 80 minutes

par Tobias Oetiker

Hubert Partl, Irene Hyna et Elisabeth Schlegl

traduit en français par Matthieu Herrb

Version 3.3
Février 1999

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

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/`

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 corrections des lecteurs du forum `fr.comp.text.tex` et en particulier de :

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

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 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* [4]. 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 pages 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 bibliographiques ou l'insertion de figures PostScript encapsulé. Il présente aussi quelques autres extensions utiles.

Le chapitre 5 contient des informations potentiellement dangeureuses. 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. Faites particulièrement attention aux exemples, 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 les sites `ftp.loria.fr` et `ftp.lip6.fr`. Aux États-Unis, il s'agit de `ctan.tug.org`, en Allemagne de `ftp.dante.de` et au Royaume-Uni de `ftp.tex.ac.uk`. Si vous n'êtes pas dans l'un de ces pays, choisissez le site le plus proche de chez vous.

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

La version courante en anglais de ce document est disponible sur :
`CTAN:/info/lshort/english`

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

Table des matières

Table des figures

Liste des tableaux

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_ε. 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.

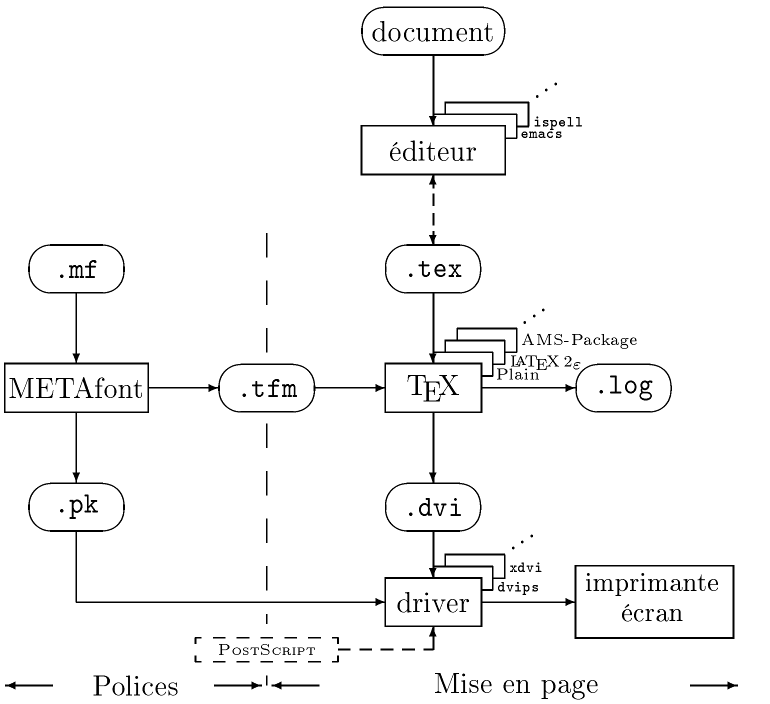


FIG. 1.1 – *Éléments d'un système TeX*

Récemment, 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 il y a quelques années. 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 **[laıtex]** et **[laıtex 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 pages 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. What you see is what you get – Ce que vous voyez est ce qui sera imprimé.

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 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] ;

- 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'autre 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.

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.

2. La rumeur dit que c'est un des points qui devrait être améliorés dans la future version LaTeX3
3. En langage typographique, *espace* est un mot féminin. *NdT.*

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. 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 `{}` empèche 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 19 octobre 2000.

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

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

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.

```
% Demonstration :
Ceci est un % mauvais
exemple.
```

Ceci est un 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}`

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

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.

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

FIG. 1.2 – *Un fichier LaTeX minimal*

```
\documentclass[a4paper,11pt]{article}
\usepackage{latexsym}
\author{H.~Partl}
\title{Minimalism}
\frenchspacing
\begin{document}
\maketitle
\tableofcontents
\section{Start}
Well and here begins my lovely article.
\section{End}
\ldots{} and here it ends.
\end{document}
```

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

1.5 La mise en page du document

1.5.1 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 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 2_ε 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,...

`report` pour des rapports plus longs contenant plusieurs chapitres, des petits livres, des thèses,...

`book` pour des vrais livres.

`slides` pour des transparents. Cette classe utilise de grands caractères sans serif.

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

```
\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*.

1.5.2 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. Un 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_ε (Reportez-vous au tableau 1.3). D'autres sont distribuées à part. Le *Local Guide* [4] 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_ε. Ce livre contient la description de centaines d'extensions ainsi que les informations nécessaires pour écrire vos propres extensions à LaTeX 2_ε.

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 connait 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.

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` et dans *The LATEX Companion* [3].

latexsym permet l'utilisation de la police des symboles LATEX.
Décrite dans `latexsym.dtx` et dans *The LATEX Companion* [3].

makeidx fournit des commandes pour réaliser un index.
Décrite dans ce document, section 4.3 et dans *The LATEX Companion* [3].

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.

1.5.3 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.

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.

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

```
\thispagestyle{style}
```

Au chapitre 4, page 61, vous apprendrez comment créer vos propres en-têtes et pieds de pages.

1.6 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{nom de fichier.tex}
```

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

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{nom de fichier,nom de 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{`*nom de fichier*`}`

qui insère simplement le fichier indiqué sans aucun traitement sophistiqué.

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 d'autant 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 un 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, notament 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 suffisament les commandes utilisées, lisez d'abord la suite du chapitre puis revenez à cette section.)

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

La loi de tensions de Kirchhof s'en déduit simplement...

% Exemple 3
... 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. ...
```

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 abbréviation. (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 ce qu'il faut faire.

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

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

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 si nécessaire. Si l'algorithme de césure[4] ne trouve pas l'endroit correct pour couper un mot, 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{An-ti-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 occurence du mot.

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

I think this is: supercalifragilisticexpialido-
cious

2. débordement horizontal
3. boîte horizontale pas assez pleine
4. *Hyphenation* en anglais

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 *exceptionnellement* 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 Caractères spéciaux et symboles

2.3.1 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. Avec LaTeX, utilisez deux `` ` `` pour les guillemets ouvrants et deux `'` pour les guillemets fermants.[5]

```
``Please press the `x' key.''

<< Appuyez sur la touche `x'. >>
```

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

2.3.2 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 :

5. En français, avec l'extension french, utilisez `<<` et `>>` *NdT*.

```
belle-fille, \\
pages 13--67\\
oui---ou non ? \\
$0$, $1$ et $-1$
```

belle-fille,
pages 13–67
oui—ou non ?
0, 1 et -1

2.3.3 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 [6]. À la place on utilise la commande spécifique :

```
\ldots
```

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

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

2.3.4 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 [7].

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

Not shelﬀul
but shelfful

2.3.5 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.

6. sauf avec l'extension `french` qui rend la commande `\ldots` inutile.
7. Je n'ai pas trouvé d'exemple en français. *NdT.*

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`
ŏ	`\u o`	ǒ	`\v o`	ő	`\H o`	ǫ	`\c o`
ọ	`\d o`	o̲	`\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`	¡	`` !` ``	¿	`` ?` ``

2.4 LaTeX international

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[8] 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* [4] devrait vous indiquer quelles sont les langues supportées par votre installation de LaTeX et comment configurer les règles de césure.
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 pour le français).

8. « Table des matières », « Liste des figures »,...

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* [4].

Pour le français, l'extension french a été développée par Bernard Gaulle [11]. En plus des règles de césure spécifiques du français, elle adapte LaTeX aux règles de la typographie française [12].

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[9]. En utilisant cette extension, il faut cependant prendre garde au fait que tous les systèmes informatiques n'utilisent pas le même codage des caractères spéciaux. D'autres personnes risquent donc de ne pas pouvoir lire votre document source, parce que leur système utilise un codage différent. Prenez-y garde si vous souhaitez échanger des documents.

Une autre difficulté provient du codage des polices de caractères utilisées par TeX, qui définit à quelle position dans une police 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}
```

9. L'utilisation du moteur TeX modifié MlTeX permet également de gérer directement les caractères spéciaux. *NdT.*

2.5 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.

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

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

1 234 567

Terminons par une subtilité de la typographie anglo-saxonne : 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\\
cf.~Fig.~5\\
I like basic\@. Do you?
```

M. Dupont vous remercie cf. Fig. 5 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 les extensions french ou babel. Dans ce cas, la commande `\@` n'est pas nécessaire.

2.6 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 :

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

Dans les classes `report` et `book`, les commandes

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

sont également reconnues, introduisant deux niveaux de structuration supplémentaires[10].

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[11] 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`). 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}
```

10. 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`.

11. Pour la classe article, elle change la numérotation des sections

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 8.

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`, `\mainmatter` et `\backmatter`

Elles permettent de séparer en trois parties votre ouvrage. Ces commandes modifient les titres des chapitres et la numérotation des pages comme cela se fait dans un livre.

2.7 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.

```
Une r\'ef\'erence \`a cette
section\label{ma-section}
ressemble \`a :
<< voir section~\ref{ma-section},
page~\pageref{ma-section}. >>
```

Une référence à cette section ressemble à : « voir section 2.7, page 25. »

2.8 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%
\footnote{ceci est une note
  de bas de page.}
sont tr\`es pris\'ees 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.

2.9 Souligner l'importance d'un mot

Dans un manuscrit réalisé sur une machine à écrire, les mots importants sont <u>soulignés</u>. Dans un ouvrage imprimé, ils sont *mis 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 ci le texte est déja en italique. Ce comportement dépend de la définition de la classe de document.

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

Pour insister *dans un passage déja 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 \`a \'ecrire}.}
```

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.

2.10 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.

2.10.1 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 \`a sa guise :
\begin{itemize}
\item mais cela peut ne pas
  \^etre  tr\`es beau,
\item ni facile \`a suivre.
\end{itemize}
\item Souvenez-vous :
\begin{description}
\item[Clart\'e :] les faits ne
vont pas devenir plus sens\'es
parce  qu'ils sont dans une liste,
\item[Synth\`ese :] cependant une
liste peut tr\`es bien
r\'e\-su\-mer 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'extension `french` utilise une présentation des listes simples qui respecte les règles typographiques françaises :

```
Une liste simple fran\c{c}aise :
\begin{itemize}
\item voici un \'el\'ement ;
\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

2.10.2 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\\
align\'e \`a 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\\
align\'e \`a 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.

2.10.3 Citations et vers

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

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

C'est pourquoi les pages
compos\'ees 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.

```
Voici le d\'ebut d'une
fugue de Boris Vian :
\begin{flushleft}
\begin{verse}
Les gens qui n'ont plus
  rien \`a 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.

2.10.4 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.

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|\ldots| \ldots

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

La commande `\ldots` …

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

```
\begin{verbatim*}
La version \'etoil\'ee de
l'environnement  verbatim
met    les   espaces   en
\'evidence
\end{verbatim*}
```

```
La␣version␣\'etoil\'ee␣de
l'environnement␣␣verbatim
met␣␣␣␣les␣␣␣espaces␣␣␣en
\'evidence
```

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 des paramètres d'autres commandes.

2.10.5 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 & hexad\'ecimal \\
3700 & octal \\
11111000000 & binaire \\
\hline \hline
1984 & d\'ecimal \\
\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

Comme il n'y a pas de commande prévue[13] pour aligner les nombres sur le point décimal (ou la virgule si on respecte les règles françaises) nous

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

pouvons « tricher » et réaliser cet alignement en utilisant deux colonnes : la 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

2.11 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 2.2.

TAB. 2.2 – *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 [14]. Quand une nouvelle page est entamée, LaTeX essaye d'abord de 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,

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

> c'est souvent à cause d'un seul objet trop grand qui bouche l'une des deux files d'objets flottants.

Après avoir rapidement expliqué cette difficulté, 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 »[15] 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]{LLLLLoooooonnnnnggggguuuueee}
```

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

Dans l'exemple ci-dessus[16] 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.

15. Avec l'extension `french`, la présentation des légendes est modifiée pour obéir aux règles françaises.
16. En supposant que la file des figures soit vide.

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.

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'une des extensions de 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 carr\'e
et $b$ au carr\'e pour obtenir
$c$ au carr\'e. 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'$\heartsuit$ 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

1. *American Mathematical Society* = Société Américaine de Mathématiques. De nombreuses extensions présentées dans ce chapitre sont développées par l'AMS.

\[et \] ou entre `\begin{displaymath}` et `\end{displaymath}`. On obtient ainsi des formules qui ne sont pas numérotées. Si vous voulez qu'elles soient numérotées par LaTeX, utilisez l'environnement `equation`.

```
Ajoutez $a$ au carr\'e
et $b$ au carr\'e pour obtenir
$c$ au carr\'e. Ou, en
utilisant une approche plus
matheuse :
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
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'\'equation (\ref{eq})
nous donne...
```

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

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}
```

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

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}
```

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

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 3.10, page 50.

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.

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

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

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. Les chapeaux et les tildes larges, couvrant plusieurs caractères, sont produits par les commandes `\widetilde` et `\widehat`. La commande `'` produit un prime.

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

```
\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}
```

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

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`

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

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

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}
```

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

Les **intégrales** sont produites par la commande `\int`, les **sommes** par la commande `\sum`. 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
\end{displaymath}
```

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

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 52.

```
\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 petite espace (␣), `\␣` pour une espace de taille moyenne (␣ représente le caractère « espace »), `\quad` (␣␣) et `\qquad` (␣␣␣) pour des espaces plus larges La commande `\!` produit une espace négative de $-3/18$ quad (␣).

```
\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}
```

$$\int\!\!\!\int_{D} g(x,y) \, \mathrm{d}x\, \mathrm{d}y$$

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`.

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 AMS-LATEX ou au chapitre 8 de *The LATEX Companion* [3]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}
```

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

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}
```

$$\begin{array}{rcll} f(x) & = & \cos x & (3.5) \\ f'(x) & = & -\sin x & (3.6) \\ \int_{0}^{x} f(y)\,dy & = & \sin x & (3.7) \end{array}$$

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}}
```

$$\begin{aligned} \sin x \;=\; & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ & - \frac{x^7}{7!} + \cdots \qquad (3.8) \end{aligned}$$

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

$$\begin{aligned} & \cos x = 1 - \frac{x^2}{2!} + \\ & \qquad + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \qquad (3.9) \end{aligned}$$

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 [10] et [3].

3.6 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` ($\displaystyle 123$), `\textstyle` ($\textstyle 123$), `\scriptstyle` ($\scriptstyle 123$) and `\scriptscriptstyle` ($\scriptscriptstyle 123$).

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}
```

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

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

3.7 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{i\`eme}}
&&\textrm{pour tout }x\\
2^{\mathrm{nd}}
&&\mathrm{pour tout }x\\
2^{\text{i\`eme}}
&&\text{pour tout }x
\end{eqnarray*}
```

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

3.8 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 montrer clairement que l'environnement `\newtheorem` facile à utiliser.

```
% definitions dans le preambule
\newtheorem{loi}{Loi}
\newtheorem{decret}[loi]%
{D\'ecret}
%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 « décret » 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 3.8.1 *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`.

3.9 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}
```

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

3.10 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`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\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`	$\ast$	`\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`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\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)	$\leadsto$	`\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`
$/$	`/`	$\backslash$	`\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*

$\dots$	\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	$\mho$	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$	\Box [a]	$\Diamond$	\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*

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\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*

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\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
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\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`	$\circledS$	`\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] et dans *The LaTeX Companion* [3].

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 *The LaTeX Companion* [3] et au *LaTeX Manual* [1].

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, idraw, 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

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. Le ratio final est de 1.0, puisqu'aucune hauteur n'est spécifiée.

Pour plus d'informations, reportez vous à [8] et [13].

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 \ldots

{\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].

Pour utiliser cette fonctionnalité, l'extension makeidx doit être chargée dans le préambule[5] 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

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`.
5. sauf avec l'extension french qui gère directement les index.

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.

4.4 En-têtes améliorés

L'extension fancyhdr[6], 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`.

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.

6. 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 entetes et pieds existant
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % espace pour le filet
\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

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{`*filename*`}`

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 [9].

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{command}
\ci{dum}
\end{command}
```

\dum

Dans cet exemple, nous utilisons à la fois un nouvel environnement appelé `command` 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 `command` 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 d' « une courte introduction à LaTeX 2_ε». Une telle commande pourrait être utile si vous aviez à citer de nombreuses fois le titre de ce livre.

```
\newcommand{\ucil}
    {Une courte (?)
     introduction \`a \LaTeXe}
% dans le document :
Voici <<\ucil>>\ldots\\
```

Voici «Une courte (?) introduction à LaTeX 2_ε» …

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 \`a \LaTeXe}
% dans le document :
\begin{itemize}
\item \uxil{courte}
\item \uxil{rapide}
\end{itemize}
```

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

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 2_ε ignore la nouvelle définition.

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, \ldots
\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 77, pour `\stretch`, voir page 71 enfin, pour plus d'informations sur `\hspace`, voir page 71.

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.

```
% 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*

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 notament 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,...). 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}
r\'egn\`erent 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.

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{...}`	`$\mathnormal{R_{19}} \neq R_{19}$`	$\mathnormal{R_{19}} \neq R_{19}$
`\mathit{...}`	`$\mathit{ffi}\neq ffi$`	$\mathit{ffi}\neq ffi$

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.

```
Il aime les  {\LARGE grands et
{\small les petits}
caract\'eres}.
```

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 :

```
{\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 \ldots
\end{Large}
```

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

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 document, 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 pr\'eambule ou dans
% une extension :
\newcommand{\danger}[1]
  {\textbf{#1}}
% dans le document :
D\'efense d'\danger{entrer}.
Cette pi\`ece 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 occurences 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 petit conseil :

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

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[1]. Bien sûr cela ne marche que si `\parindent` n'est pas nul.

Pour créer un paragraphe sans indentation, utilisez :

`\noindent`

en tête du paragraphe.

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

l'extension french modifie ici aussi les règles par défaut de LaTeX pour s'adapter aux règles françaises. Elle propose en plus la commande :

`\lettrine{`*texte*`}`

qui permet de commencer le premier paragraphe d'un document par une lettrine telle que celle de la page iii.

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, les sections, les 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.

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.*

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 \ldots

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

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

`\\[`*longueur*`]`

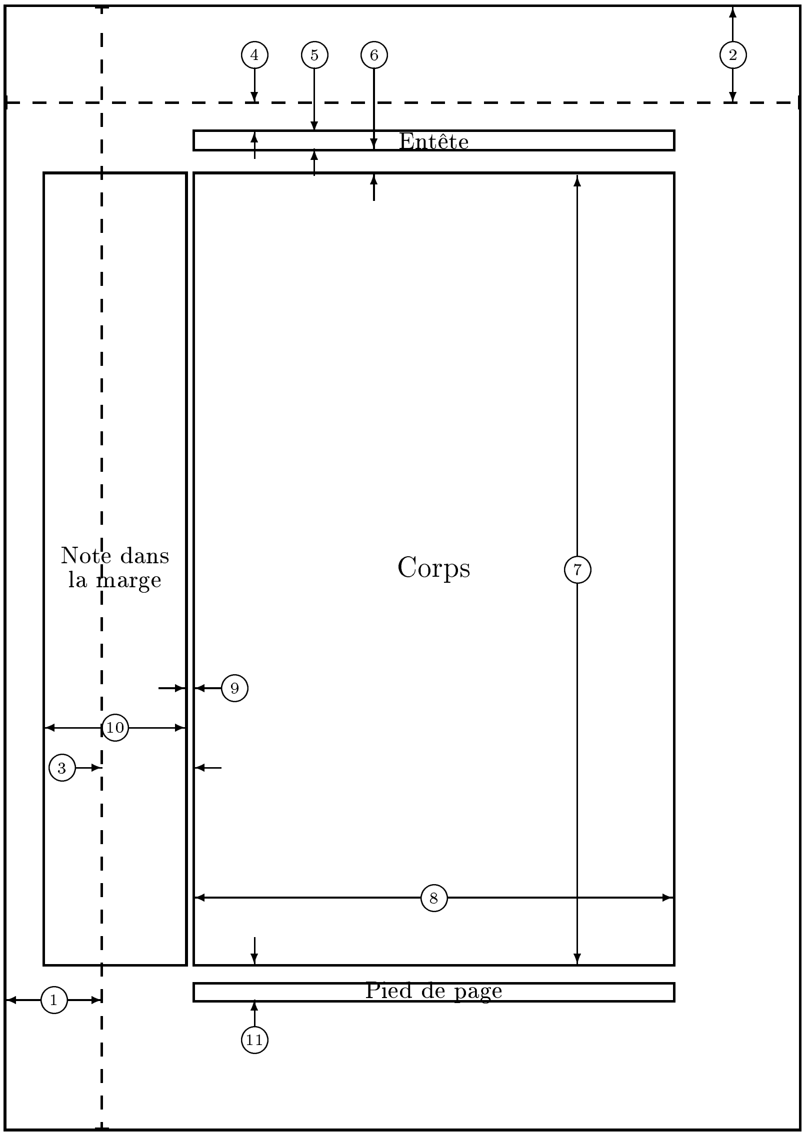
5.4 Disposition d'une page

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 raison 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[2] et comptez le nombre de caractères sur une ligne normale. Vous verrez qu'il

2. 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*

n'y a guère plus de soixante-six caractères par ligne. L'expérience montre que les 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 contôlent la disposition d'une page, voici comment procéder :

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

La première commande affecte une valeur fixe à l'un quelconque de ces paramètres :

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

La deuxième commande ajoute une longueur à l'un des paramètres.

`\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}{O\`u :}$a$,
$b$ sont les cot\'es adjacents \`a
l'angle droit d'un triangle
rectangle,\par
$c$ est l'hypoth\'enuse du
triangle,\par
$d$ n'est pas utilis\'e 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'alignement 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[3]. 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]{%
\'e t i r \'e}\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 ?
```

c e n t r é

é t i r é

Whoua le cadre !

Raté, je suis trop large

c'est aussi mon cas. Pouvez-vous lire ceci ?

3. 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.

Maintenant que nous savons contrôler l'alignement horizontal, la suite logique est de voir comment gérer l'alignement vertical[4]. Pas de problème 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 \'etait arriv\'ee.
```

Aaaaaaargh cria-t-il, mais la ligne suivante ne remarqua pas qu'une chose horrible lui était arrivée.

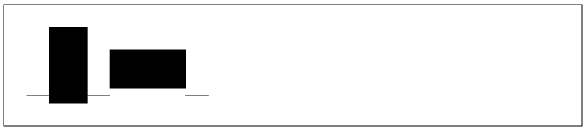
5.7 Filets

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.

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

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}\'Etai\ldots\\
\hline
\rule{0pt}{4ex} montant \\
\hline
\end{tabular}
```

Étai ...
montant

Bibliographie

[1] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.

[2] Donald E. Knuth. *The TeXbook,* Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach et Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.

[4] 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.

[5] LaTeX3 Project Team. *LaTeX 2ε for authors.* Distribué avec LaTeX 2ε dans `usrguide.tex`.

[6] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers.* Distribué avec LaTeX 2ε dans `clsguide.tex`.

[7] LaTeX3 Project Team. *LaTeX 2ε Font selection.* Distribué avec LaTeX 2ε dans `fntguide.tex`.

[8] David P. Carlisle. *Packages in the 'graphics' bundle.* Distribué avec les extensions « graphics » dans `grfguide.tex`.

[9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of LaTeX's verbatim Environments.* Distribué avec l'ensemble « tools » dans `verbatim.dtx`.

[10] American Mathematical Society *AMS-LaTeX Version 1.2 User's guide.* Distribué avec les extensions AMS-LaTeX dans `amsldoc.tex`.

[11] Bernard Gaulle. *Notice d'utilisation du style french multilingue.* Disponible avec le style `french` sur `ftp.univ-rennes1.fr:/pub/GUTenberg/french`.

[12] Yves Perrousseaux. *Manuel de typographie française élémentaire.* Ateliers Perrousseaux éditeur, 1995, ISBN 2-911220-00-5.

[13] Keith Reckdahl. *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`