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

ou LaTeX2e en 73 minutes

par Tobias Oetiker

Hubert Partl, Irene Hyna et Elisabeth Schlegl

traduit en français par Matthieu Herrb

Version 1.0
Juillet 1996

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:/tex-archive/info/lkurz`

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

La version courante en français est disponible sur :
`ftp.laas.fr:/pub/ii/logiciels/latex/flshort` en attendant d'être archivée sur CTAN.

Pour la préparation de ce document, l'aide des lecteurs du forum USENET `comp.text.tex` a été sollicité. 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, Chris McCormack,
David Dureisseix, Elliot, Robin Fairbairns,
Alexandre Guimond, Cyril Goutte, Neil Hammond,
Rasmus Borup Hansen, Martien Hulsen, Eric Jacoboni,

Alan Jeffrey, Byron Jones, David Jones, Andrzej Kawalec,
Christian Kern, Jörg Knappen, Maik Lehradt, Claus Malten,
Hubert Partl, John Refling, Mike Ressler, Brian Ripley,
Young U. Ryu, Chris Rowley, Craig Schlenter, and
Josef Tkadlec.

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

Vincent Zoonekynd

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

Préface

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

Cette introduction décrit LaTeX 2$_\varepsilon$ 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, du PC d'IBM aux plus gros CRAY. 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 quatre chapitres :

Le chapitre 1 présente la structure élémentaire d'un document LaTeX 2$_\varepsilon$. 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 du processus de formattage d'un document. Il explique les commandes et les environnements essentiels de LaTeX. Après avoir lu ce chapitre, vous serez capables de rédiger votre premier document.

Le chapitre 3 explique comment formatter des formules mathématiques avec LaTeX. De nombreux exemples y sont donnés pour mettre en valeur cet atout majeur de LaTeX. À la fin de ce chapitre, vous trouverez des tableaux qui listent tous les symboles mathématiques disponibles dans LaTeX.

Le chapitre 4 présente enfin divers outils de LaTeX qui ne sont pas indispensables, mais cependant bien pratiques. Entre autres choses, vous apprendrez à insérer des figures PostScript dans vos documents et à gérer un index et une liste de références bibliographiques.

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.ibp.fr`. Aux États-Unis, il s'agit de `ftp.shsu.edu`, 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 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, Zurich.

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

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.

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 pages.

Récemment, le package 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 appeléeLaTeX 2_ε. Ce document est relatif à LaTeX 2_ε.

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

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

1.2 Les Bases

1.2.1 Auteur, éditeur, et typographe

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

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

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

C'est une approche totalement différent de l'approche WYSIWYG[1] utilisée par les traitement de textes modernes tels que *Word pour Windows* ou *FrameMaker*. Avec ces programmes, l'auteur définit la mise en pages du document de manière interactive pendant la saisie du texte. Tout au long de cette opération, il voit à l'écran à quoi ressemblera le document final une fois imprimé.

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

1.2.2 Choix de la mise en pages

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.

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

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 formattage 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...

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.
- composition des formules mathématiques de manière pratique.
- il est seulement nécessaire 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 supplémentaires gratuites. Par exemple pour inclure des figures PostScript ou pour formatter 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 formattage 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 consomme plus de ressources (mémoire, espace disque, puissance de calcul) qu'un traitement de texte normal. Mais les choses changent, puisque Word pour Windows version 6.0 consomme plus de place disque qu'un système LaTeX 2_ε normal. En ce qui concerne la puissance de calcul, LaTeX bat tous les traitement de texte WYSIWYG, puisqu'il ne consomme de temps CPU que durant le formattage, alors qu'un logiciel WYSIWYG recalcule celle-ci après chaque frappe de caractère.

– 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].

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 *un seul* espace. Les espaces en début d'une ligne sont en général ignorés et un seul retour à la ligne est traité comme un espace.

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

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

Une ligne vide commence un nouveau paragraphe.

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

Une ligne vide commence
un nouveau paragraphe.
```

1.3.2 Caractères spéciaux

Les symboles suivants sont des caractères réservés qui, soit ont une signification spéciale dans LaTeX, soit ne sont pas disponibles dans toutes les polices. Si vous les saisissez directement dans votre texte, ils ne seront

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

pas imprimés mais 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 document en les préfixant par un antislash :

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

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

1.3.3 Commandes LaTeX

Les commandes LaTeX sont sensibles à la casse des caractères (majuscules / 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 un 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 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 TeXniciens et en TeXperts.
Ajourd'hui nous sommes le 25 juillet 1996.

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

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

You can *lean* on me!

```
You can \textsl{lean} on me!
```

Please, start a new line right here!
Thank you!

```
Please, start a new line
right here!\linebreak[3]
Thank you!
```

1.3.4 Commentaires

Quand LaTeX rencontre un caractère % en analysant 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.

C'est un exemple pour tout le monde

```
C'est  un  exemple % stupide
% Mieux: instructif <----
pour tout le monde
```

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 [3], vous pouvez commencer le corps du texte avec la commande :

```
\begin{document}
```

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

```
\end{document}
```

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

La figure 1.1 montre le contenu d'un document LaTeX 2_ε minimum. Un fichier source plus complexe est présenté sur la figure 1.2.

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

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

FIG. 1.1 – *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. 1.2 – *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 fourni d'autres classes pour d'autres types de documents, notament 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.

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

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

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

Tab. 1.1 – *Classes de document*

`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.

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 formatter 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 chapitre sur la prochaine page vierge alors que la classe `book` les commence toujours sur une page de droite.

1.5.2 Extensions

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` pour accéder à la police des symboles LaTeX, utilisez cette extension.
Décrite dans `latexsym.dtx` et dans *The LaTeX Companion* [3].

`makeidx` fournit des commandes pour réaliser un index.
Décrite dans ce document, section 4.5 et dans *The LaTeX Companion* [3].

`syntonly` analyse un document sans le formatter.
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 formatter avec la commande `latex doc.dtx`. Il en est de même pour les autres fichiers cités dans ce tableau.

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 LaTeXgrâce 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.

1.5.3 Styles de page

LaTeX propose trois combinaisons d'en-têtes et de pieds de page, appelées styles de page et définis 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*`}`

Dans *The LaTeX Companion* [3], on vous explique comment créer vos propres en-têtes et pieds de pages.

1.6 Gros documents

Lorsque l'on travaille sur des 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*`}`

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, seuls 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 formattage 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.

Attention : certaines extensions redéfinissent parfois certains caractères spéciaux[4]. Ils ne peuvent plus alors être utilisés dans les *noms de fichier*.

4. Par exemple le caractère « souligné » : _

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 Sauts de ligne et de page

2.1.1 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 indentée et il n'y a pas d'espace vertical particulier entre deux paragraphes (Cf. section 4.2.2).

Dans certains cas particuliers, il peut être nécessaire de demander à

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

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. 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.

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 protester sans arrêt.

2.1.2 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.

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

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 composé 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 ». Aucun caractère spécial ou symbole ne sont autorisés dans les arguments.

Exemple:

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

La commande `\-` insère un point de césure potentiel dans un mot. Cette position devient alors le seul point de césure possible pour ce mot. Cette commande est surtout utile pour les mots contenant des caractères spéciaux. En effet LaTeX ne fait pas la césure des mots contenant des caractère spéciaux. La plupart des versions américaines de TeX considèrent les accents comme des caractères spéciaux. Une version assez répandue hors des États-Unis, MlTeX [5], supprime cette restriction gênante pour les langues européennes. L'extension **inputenc** avec le codage T1 et la nouvelle famille de polices DC aboutissent au même résultat pour les versions récentes de LaTeX 2_ε.

I think this is: supercalifragilisticex-
pialidocious

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

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

`\mbox{`*texte*`}`

Elle a pour effet de lier ses arguments en toutes circonstances.

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.

```
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.
```

5. Développée par Michael J. Ferguson et Bernd Raichle

2.2 Caractères spéciaux et symboles

2.2.1 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.[6]

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

2.2.2 Tirets

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

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

2.2.3 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 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[7] À la place on utilise la commande spécifique :

```
\ldots
```

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

6. En français, avec l'extension french, utilisez `<<` et `>>` *NdT.*
7. sauf avec l'extension french qui rend la commande `\ldots` inutile.

2.2.4 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 des mots composés[8]

Not shelfful
but shelfful

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

2.2.5 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.

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

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

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

Tab. 2.1 – *Accents et caractères spéciaux*

ò	`\`o`	ó	`\'o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`		
ŏ	`\u o`	ǒ	`\v o`	ő	`\H o`	o̧	`\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`	¡	`` !` ``	¿	`` ?` ``

8. Je n'ai pas trouvé d'exemple en français. *NdT.*

2.3 LaTeX international

Pour composer des documents dans des langues autres que l'anglais, LaTeX doit pouvoir utiliser des règles de césure différentes selon les langues.

Pour de nombreuses langues, ces règles différentes peuvent être obtenues en utilisant l'extension babel maintenue par Johannes Braams.

Si votre système est configuré pour supporter cette extension, vous pouvez sélectionner la langue à utiliser 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 certaines langues babel définit également de nouvelles commandes qui simplifient la saisie des caractères spéciaux.

Pour le français, il existe une extension spécifique : french, développé par Bernard Gaulle [9]. En plus des règles de césure spécifiques du français, il adapte LaTeX aux règles de la typographie française [10].

Certains systèmes informatiques vous permettent de saisir directement les caractères accentués ou les symboles spécifiques d'une langue (Soit par des touches spéciales, soit par des combinaisons programmées). LaTeX 2_ε sait gérer plusieurs codages de ces caractères spéciaux 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. Il est nécessaire d'y prendre garde si l'on souhaite échanger ses documents.

Terminons ce paragraphe en notant qu'il existe plusieurs logiciels capables de convertir le codage des accents d'un document source d'un système à l'autre. Pour échanger des documents sur *internet*, il est conseillé d'utiliser le codage standard TeX (`\'e`) qui est transmis sans risque d'erreur entre tous les systèmes.

2.4 L'espace entre les mots

Pour obtenir une marge droite alignée, LaTeX insère des espaces plus ou moins larges entre les mots. Après la ponctuation finale d'une phrase, les règles de la typographie anglo-saxonne veulent que l'on insère un espace plus large. LaTeX obéit par défaut à cette règle.

Un antislash avant un espace produit un espace qui ne peut pas être élargi. Un tilde « ~ » produit un espace qui ne peut ni être élargi, si provoquer

9. MlTeX est une autre possibilité pour gérer directement les caractères spéciaux. *NdT.*

de saut de ligne. La commande `\@` avant un point indique que celui-ci termine une phrase, même lorsqu'il suit une lettre majuscule (par défaut LaTeX considère qu'un point après une majuscule ne termine pas une phrase).

M. Dupont vous remercie cf. Fig. 5 J'aime le BASIC. Et vous?	`M.~Dupont  vous remercie\\` `cf.~Fig.~5\\` `J'aime le BASIC\@. Et vous ?`

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

```
\frenchspacing
```

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

Enfin la commande `\,` permet d'insérer un demi-espace insécable.

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

2.5 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 deux commandes supplémentaires sont reconnues :

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

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. 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[10] 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
```

est remplacée par la table des matières. Un document doit être traité par LaTeX (« LaTeXé ») deux fois 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. La commande `\section{Introdution}` pourra devenir `\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 rentrer un titre trop long dans la table des matières. On peut donner un titre spécifique pour la table des matières en argument optionnel avant le titre principal :

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

Le titre du document est obtenu par la commande :

```
\maketitle
```

Le titre lui-même est défini 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`.

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

Vous trouverez un exemple des commandes citées ci-dessus sur la figure 1.2, page 7.

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 numérotent les pages comme cela se fait dans un livre.

2.6 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. Ici également, les numéros sont issus d'une exécution précédente de LaTeX (stockés dans le fichier `.aux`).

Une référence à cette section ressemble à : « voir section 2.6, page 21. »

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

2.7 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[a] sont très prisées par les utilisateurs de LaTeX.

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

```
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{}.
```

2.8 Souligner l'importance d'un mot

Dans un manuscrit réalisé sur une machine à écrire, les mots importants sont souligné. Dans un ouvrage imprimé, ils sont *mis en valeur*[11]. La commande mettre en valeur un passage est :

`\emph{`*texte*`}`

Son argument est le texte à mettre en valeur. En général, la police *italique* est utilisée pour la mise en valeur, mais ceci dépend de la définition de la classe de document.

Si vous voulez souligner l'importance *d'un mot dans un passage déja mis en valeur, LaTeX utilise alors une police droite.*

```
\emph{Si vous voulez
  \emph{souligner l'importance}
d'un mot dans un passage d\'eja
mis en valeur, \LaTeX{} utilise
alors une police droite.}
```

2.9 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.

11. *Emphasised* en anglais.

2.9.1 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.

1. vous pouvez mélanger les environnements à votre 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.

```
\begin{enumerate}
\item vous pouvez m\'elanger les
environnements \`a votre 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}
```

Notez que l'extension french utilise une présentation des listes simples qui respecte les règles typographiques françaises :

Une liste simple française :

– voici un élément

– puis un autre.

```
Une liste simple fran\c{c}aise :
\begin{itemize}
\item voici un \'el\'ement
\item puis un autre.
\end{itemize}
```

An english list:

- one item
- an other one

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

2.9.2 Alignements à gauche, à droite et centrage

Les environnements `flushleft` et `flushright` produisent des paragraphes 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.

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

```
\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 droite. LaTeX n'essaye pas
d'aligner la marge gauche.

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

Au centre de la terre.

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

2.9.3 Citations et vers

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

Une règle typographique simple pour la longueur des lignes :

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

Cela explique pourquoi les journaux utilisent souvent plusieurs colonnes.

```
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.
```

Il existe deux autres environnements comparables : `quotation` et `verse`. L'environnement `quotation` est utile pour des citations plus longues, couvrant plusieurs 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 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.

```
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}
```

2.9.4 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 `\ldots` …

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

```
La commande \verb|\ldots| \ldots

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

```
La␣version␣\`etoile\`e␣de
l'environnement␣␣verbatim
met␣␣␣␣les␣␣␣espaces␣␣␣en
\'evidence
```

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

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

`comme␣ceci␣:-)␣`

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

L'environnement `verbatim` et la commande `\verb` ne peuvent être utilisés à l'intérieur des paramètres d'autres commandes.

2.9.5 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 de texte aligné à gauche, `r` pour du texte aligné à droite et `c` pour du texte centré. `p{`*largeur*`}` permet de réaliser une colonne de texte justifié à droite sur plusieurs lignes, et enfin `|` permet d'obtenir une ligne verticale.

À l'intérieur de l'environnement `tabular`, le caractère `&` passe à la colonne suivante, `\\` commence une nouvelle ligne et `\hline` insère une ligne horizontale.

7C0	hexadécimal
3700	octal
11111000000	binaire
1984	décimal

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexad\'ecimal \\
3700 & octal \\
11111000000 & binaire \\
\hline \hline
1984 & d\'ecimal \\
\hline
\end{tabular}
```

Bienvenue dans ce cadre. Merci de votre visite.

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

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 au sujet du problème de l'alignement des nombres décimaux. Une autre utilisation possible est de supprimer l'espacement dans un tableau avec `@{}`.

sans espace

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

avec espaces

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

Comme il n'y a pas de commande prévue[12] pour aligner les nombres sur le point décimal (ou la virgule si on respecte les règles françaises) nous pouvons « tricher » et réaliser cet alignement en utilisant deux colonnes : la première alignée à droite contient la partie entière et la seconde alignée à gauche contient la partie décimal. La commande `\@{.}` dans la description du tableau remplace l'espace normal entre les colonnes par un simple « . », donnant l'impression d'une seule colonne alignée sur le point. N'oubliez pas de remplacer dans votre tableau le point par un séparateur de colonnes (`&`) ! Un label peut être placé au-dessus de cette colonne en utilisant la commande `\multicolumn`.

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

```
\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}
```

2.10 Objets dérivants

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 approche

12. Si les extensions de l'ensemble « tools » sont installées sur votre système, jetez un œil sur l'extension dcolumn.

laisserait de nombreuses pages à moitié blanches, ce qui ne serait réellement pas beau.

La solution est de laisser « dériver » les figures et les tableaux qui ne rentrent pas sur la page en cours, vers une page suivante et de terminer de remplir la page avec le texte. LaTeX fournit deux environnements pour les objets dérivants 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 dérivants de manière interne. Sinon ces objets deviennent 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 dérivants :

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

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

appelé *placement*. Ce paramètre permet de dire à LaTeX où vous autorisez l'objet à dériver. Un *placement* est composé d'une chaîne de caractères représentant des *placements possibles*. Reportez-vous au tableau 2.2.

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.

TAB. 2.2 – *Placements possibles*

Caractère	Emplacement pour l'objet dérivant ...
`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 a part ne contenant que des objets dérivants.
`!`	ici, sans prendre en compte les paramètres internes[a] qui pourraient empêcher ce placement.

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

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[13]. 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 dérivant qui ne peut être placé dans le texte repousse toutes les autres à la fin du document.

Ainsi :

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

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{`*caption text*`}`

vous définissez une légende pour l'objet. Un numéro (incrémenté automatiquement) et le mot « Figure » ou « Table »[14] 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.

13. Il s'agit de files FIFO (*First In, First Out*).

14. Avec l'extension `french`, la présentation des légendes est modifiée pour obéir aux règles françaises.

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.

```
Figure~\ref{blanche} est un exemple de Pop-Art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Cinq centimètre sur cinq} \label{blanc}
\end{figure}
```

Dans l'exemple ci-dessus[15] LaTeX va s'acharner (`!`) à placer la figure là où se trouve la commande (`h`) dans le texte. S'il n'y arrive pas, il essayera de la placer en bas (`b`) de la page. Enfin s'il ne peut la placer sur la page courante, il essayera de créer une page à part avec d'autres objets flottants. S'il n'y a pas suffisamment de tableaux en attente pour remplir une page spécifique, LaTeX continue et, au début de la page suivante, réessayera de placer la figure comme si elle venait d'apparaître dans le texte.

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

`\clearpage` ou même `\cleardoublepage`

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

À la section 4.6, vous apprendrez à inclure une figure PostScript dans votre document LaTeX 2_ε.

2.11 Nouvelles commandes et nouveaux environnements

Dans le premier chapitre, nous avons vu que LaTeX avait besoin de connaître la structure logique d'un document pour pouvoir le formatter correctement. Ceci est une idée intéressante, mais en pratique on atteint assez vite ses limites quand on ne trouve pas dans LaTeX la commande spécialisée pour exprimer ce que l'on voudrait.

Une solution serait de combiner plusieurs commandes LaTeX pour obtenir le résultat désiré. Ponctuellement, cela ne pose pas de problème ; mais si cela se reproduit, répéter chaque fois la même combinaison peut devenir long et fastidieux. De plus si vous souhaitez plus tard modifier la présentation, il faudra aller modifier chaque combinaison de commandes tout au long du texte.

Pour résoudre ce problème, LaTeX vous permet de définir vos propres commandes et vos propres environnements.

15. En supposant que la file des figure soit vide.

2.11.1 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 2ε ». Une telle commande pourrait être utile si vous aviez à citer de nombreuses fois le titre de ce livre.

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

```
\newcommand{\ucil}
    {Une courte (?)
     introduction \'a \LaTeXe}
% dans le document :
^^ab \ucil ^^bb \ldots{}
 ^^ab \ucil ^^bb
```

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

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

```
\newcommand{\uil}[1]
    {Une \emph{#1}
       introduction \'a \LaTeXe}
% dans le document :
\begin{itemize}
\item \uil{courte}
\item \uil{petite}
\end{itemize}
```

LaTeX ne vous permet pas de créer une nouvelle commande si celle-ci existe déjà. Si vous voulez explicitement remplacer une commande existante, utilisez `\renewcommand`. Elle utilise la même syntaxe que `\newcommand`. Dans certains cas, vous pouvez avoir besoin de `\providecommand`. Cette commande fonctionne comme `\newcommand`, mais si la nouvelle commande est déjà définie, LaTeX 2ε ignore la nouvelle définition.

2.11.2 Nouveaux environnements

De manière analogue à la commande `\newcommand`, il existe une commande pour définir de nouveaux environnements. Pour rédiger ce document, elle a été utilisée plusieurs fois pour définir des environnements utilisés dans cette présentation : « exemples », « morceaux de code » et « boites de définition de commandes ». 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`.

Vive le Québec libre !

```
\newenvironment{degaulle}
    {\begin{quote}}{\end{quote}}
% dans le texte :
\begin{degaulle}
Vive le Qu^^e9bec libre !
\end{degaulle}
```

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`.

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 [1]-LATEX.

3.1 Généralités

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

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

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

100 m^3 of water

This comes from my $\heartsuit$

```
100~m$^{3}$ of water\\[6pt]
This comes from my $\heartsuit$
```

Il vaut mieux exposer les équations ou les formules plus importantes, c'est-à-dire les composer sur des lignes à part. Pour cela, on les inclut entre `\[` et `\]` 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`.

1. *American Math Society.* De nombreuses extensions présentés dans ce chapitre sont développés par l'AMS.

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

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

Le mot de la fin.

```
Ajoutez $a$ au 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.
```

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

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

L'équation (3.1) nous donne...

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

Remarquez que les expressions mathématiques sont formattées différemment dans un paragraphe ou quand elles sont exposées. Comparez :

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

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

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

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 déduits de la logique de la formule ou indiqués à l'aide de commandes spécifiques telles que : `\,`, `\quad` ou`\qquad`.

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

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

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{...}`.

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

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

Les mathématiciens sont très tatillons au sujet des symboles à utiliser. Il serait plus conventionnel ici d'utiliser la police « blackboard bold » qui est obtenue par la commande `\mathbb` de l'extension amsfonts ou amssymb. L'exemple précédent devient :

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

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

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 : `{...}`.

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

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

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 45.

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

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

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

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

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

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

```
$a_{1}$ \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`.

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

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

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.

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

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

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

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

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

Les noms des fonctions telles que log sont souvent 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

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

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

Pour la fonction modulo, il y a deux commandes : `\bmod` pour l'opérateur binaire « $a \bmod b$ » et `\pmod` pour les expressions telles que « $x \equiv a \pmod{b}$ ».

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

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

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

Pour imprimer des coefficients binomiaux ou d'autres structures semblables, utilisez la commande `\binom` de l'extension amsmath :

$$\binom{n}{k}\qquad \dbinom{x}{y+2}$$

```
\begin{displaymath}
\binom{n}{k}\qquad \dbinom{x}{y+2}
\end{displaymath}
```

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.

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

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

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 correspondant, 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 47.

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

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

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.`!

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

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

Dans certains cas, il est nécessaire d'indiquer la taille exacte des délimiteurs matémathiques 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.

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

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

3. Ces commandes ne fonctionnent pas correctement après une commande de changement de taille ou si les options `11pt` ou `12pt` sont utilisés. 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 un petit espace, `\␣` pour un espace de taille moyenne (␣ représente le caractère « espace »), `\quad` et `\qquad` pour des espaces plus larges et `\!` qui réduit un espace.

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

plutôt que

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

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
  \, \ud x\, \ud y
\end{displaymath}
plut^^f4t que
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

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

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.

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

```
\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}
```

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

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

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{si $d>c$}\\
 b+x & \textrm{le matin}\\
 l & \textrm{la journ^^e9e}
  \end{array} \right.
\end{displaymath}
```

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

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.

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

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

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.

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

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

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

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

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

Il peut être difficile d'obtenir des équations alignées verticalement correctement avec cette méthode. L'extension amsmath fournit d'autres possibilités plus puissantes.

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. Pour ajouter du texte normal dans une équation, et que vous utilisez la commande `\textrm`, le mécanisme d'ajustage de la taille ne fonctionnera pas. En effet `\textrm` bascule provisoirement en mode texte. Utilisez plutôt la commande `\mathrm` qui laisse le mécanisme d'ajustement de la taille des caractères actif. Mais attention, `\mathrm` ne marche que pour des éléments relativement courts. Les espaces ne sont pas actifs et les accents ne marchent pas[4].

$$2^{\mathrm{nd}} \tag{3.10}$$

```
\begin{equation}
2^{\mathrm{nd}}
\end{equation}
```

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.

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

```
\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}
```

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 Décrire des variables

Après certaines formules, on souhaite parfois ajouter une section qui décrit les variables utilisées dans l'expression. L'exemple suivant devrait vous

4. l'extension amsmath fournit une commande `\text` qui marche avec les ajustements de taille.

aider à réaliser cela.

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

Où : a, b – sont les cotés adajacents à l'angle droit du triangle rectangle.

c – est l'hypoténuse du triangle.

```
\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}
{\settowidth{\parindent}
   {Ou :\ }

\makebox[0pt][r]
 {O\`u :\ }$a$, $b$ -- sont
les cot\'es adajacents \`a
l'angle droit du triangle
rectangle.

$c$ -- est l'hypot\'enuse
du triangle.}
```

Si vous utilisez souvent des constructions telles que celle-ci, il est temps d'essayer la commande `\newenvironnement` pour créer un environnement dédié à la description des variables.

3.8 Théorèmes, Lois, ...

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. Assez de théorie. Les exemples qui suivent devraient lever les derniers doutes et montrer clairement que l'environnement `\newtheorem` est trop difficile à comprendre.

Loi 1 *Le chef a raison*

Commandement 2 (Important) *Le chef a toujours raison*

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

```
% definitions dans le preambule
\newtheorem{loi}{Loi}
\newtheorem{commandement}[loi]%
{Commandement}
%dans le document
\begin{loi} \label{chef}
Le chef a raison
\end{loi}
\begin{commandement}[Important]
Le chef a toujours raison
\end{commandement}
\begin{loi}
Si le chef a tort, voir
la loi~\ref{chef}
\end{loi}
```

Le théorème « commandement » 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 quelquechose de ce genre pour le théorème.

Murphy 3.8.1 *Tout ce qui peut aller mal, ira mal.*

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

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.

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

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

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

L'extension amsmath permet d'utiliser le gras pour un élément seulement dans une formule. Elle fournit la commande `\boldsymbol` et un « gras du pauvre[5] » `\pmb` qui agit par superposition sur les systèmes où ne sont pas installées les polices nécessaires pour les symboles gras.

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

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

5. *Poor man's bold* en anglais. *NdT.*

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[6], 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:/tex-archive/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`		

6. 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 prefixant 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*

<table>
<tr><td>$($</td><td><code>(</code></td><td>$)$</td><td><code>)</code></td><td>$\uparrow$</td><td><code>\uparrow</code></td><td>$\Uparrow$</td><td><code>\Uparrow</code></td></tr>
<tr><td>$[$</td><td><code>[</code> ou <code>\lbrack</code></td><td>$]$</td><td><code>]</code> ou <code>\rbrack</code></td><td>$\downarrow$</td><td><code>\downarrow</code></td><td>$\Downarrow$</td><td><code>\Downarrow</code></td></tr>
<tr><td>$\{$</td><td><code>\{</code> ou <code>\lbrace</code></td><td>$\}$</td><td><code>\}</code> ou <code>\rbrace</code></td><td>$\updownarrow$</td><td><code>\updownarrow</code></td><td>$\Updownarrow$</td><td><code>\Updownarrow</code></td></tr>
<tr><td>$\langle$</td><td><code>\langle</code></td><td>$\rangle$</td><td><code>\rangle</code></td><td>$\vert$</td><td><code>|</code> ou <code>\vert</code></td><td>$\Vert$</td><td><code>\|</code> ou <code>\Vert</code></td></tr>
<tr><td>$\lfloor$</td><td><code>\lfloor</code></td><td>$\rfloor$</td><td><code>\rfloor</code></td><td>$\lceil$</td><td><code>\lceil</code></td><td>$\rceil$</td><td><code>\rceil</code></td></tr>
<tr><td>$/$</td><td><code>/</code></td><td>$\backslash$</td><td><code>\backslash</code></td><td></td><td>. (dual. vide)</td><td></td><td></td></tr>
</table>

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}`	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

Spécialités

Ne lisez pas ce chapitre ! Si vous vous sentez à l'aise, vous pouvez écrire vos premiers documents avec LaTeX dès maintenant. Le but de ce chapitre est d'ajouter un peu de piment à votre connaissance de LaTeX. Une description bien plus complète des possibilités et des améliorations possibles se trouve dans le *LaTeX Manual* [1] et dans *The LaTeX Companion* [3].

4.1 Polices et tailles

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 4.1 et 4.2. La taille exacte de chaque police est un choix qui dépend de la classe de document et de ses options.

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

```
{\small Les romains
petits et \textbf{gras}
r\`egn\`erent sur}
{\Large toute la belle
et grande \textit{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. Cela peut paraître évident à quelqu'un qui débute avec LaTeX 2_ε, mais ça ne l'est certainement pas pour quelqu'un qui était habitué à LaTeX 2.09.

En mode *mathématique*, vous pouvez utiliser les commandes de changement de police pour quitter provisoirement le mode math 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 4.3.

TAB. 4.1 – *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. 4.2 – *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. 4.3 – *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 lien avec les commandes de changement de taille, les accolades jouent un rôle important. Elles sont utilisées pour construire des *groupes.* Les groupes limitent la portée de la plupart des commandes LaTeX.

Il aime les grands et les petits caractères.

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

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 :

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

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

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

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

Pour conclure cette promenade au pays des commandes de changement de police, voici un petit conseil[1] :

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

4.2 Espacement

4.2.1 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.

Quand on effectue ce changement, il est souvent indésirable dans les notes de bas de page, les tableaux, etc. C'est assez compliqué à gérer, mais l'extension setspace est prévue à cet effet.

1. Attention, c'est du second degré, j'espère que vous l'aurez compris.

4.2.2 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 augmentez l'espace entre deux paragraphes et vous supprimez l'indentation initiale. Attention, cela a également une influence sur la table des matières.

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

```
\indent
```

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

Pour créer un paragraphe sans indentation, utilisez :

```
\noindent
```

en tête du paragraphe.

l'extension french modifie ici aussi les règles par défaut de LaTeX pour s'adapter aux règles française. 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.

4.2.3 Espacement horizontal

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

```
\hspace{longueur}
```

Si un tel espace doit être conservé, même s'il 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 4.4.

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

TAB. 4.4 – *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.*

La commande :

`\stretch{`*n*`}`

produit un espace élastique. Il 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'elasticité n.

x x x

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

4.2.4 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. 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
```

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

```
\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*`]`

4.3 Disposition d'une page

LaTeX 2ε 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 4.1 montre tous les paramètres qui peuvent être modifiés. Cette figure a été réalisée avec l'extension layout de l'ensemble tools.

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.

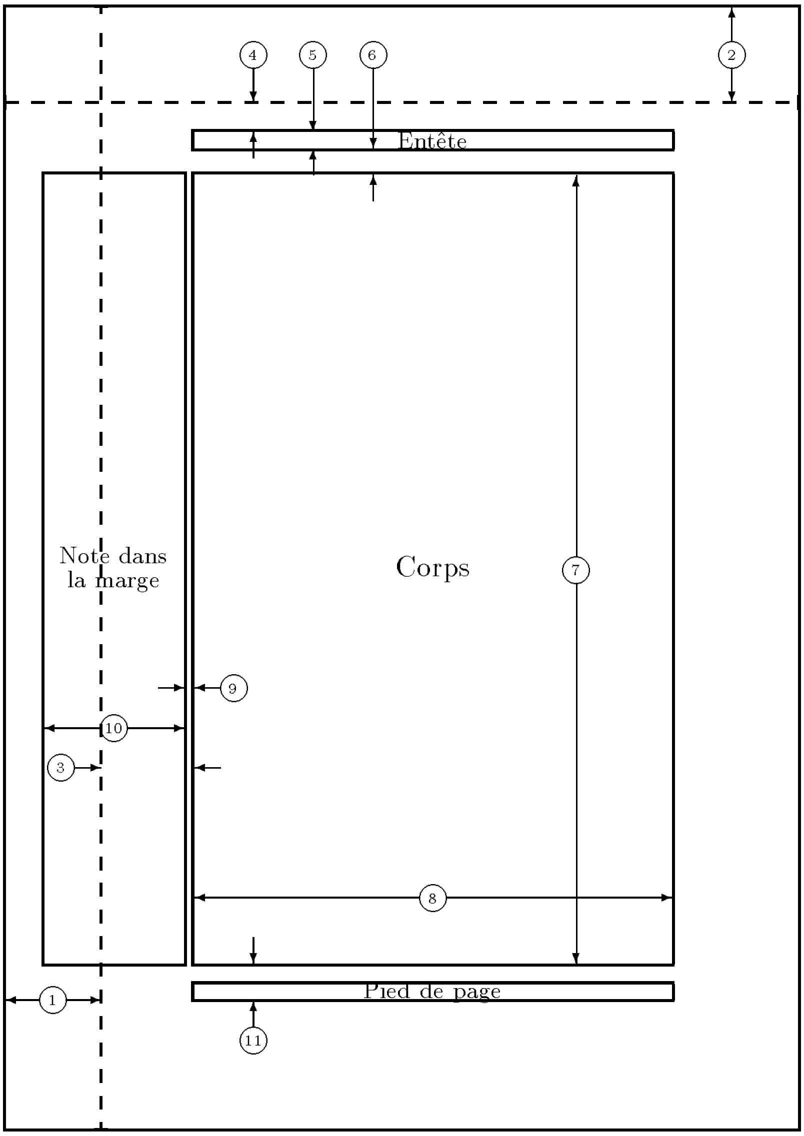
`\addtolength{`*paramètre*`}{`*longueur*`}`

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}
```

4.4 Références bibliographiques

L'environnement `thebibliography` permet de produire une liste de références bibliographiques. Cet environnement est conçu pour être produit



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. 4.1 – *Paramètres de la disposition d'une page*

automatiquement par BiBTEX. Il est cependant possible de construire de petites bibliographies à la main. 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 `\thebibliography` définit la largeur maximale de ces numéros.

Partl [1] propose que ...

Bibliographie

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

```
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}
}
```

4.5 Index

L'index est une caractéristique fort utile de nombreux ouvrages. LATEX et le programme associé `makeindex`[3] 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 chargé dans le préambule[4] avec :

```
\usepackage{makeidx}
```

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

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

TAB. 4.5 – *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 formatté
`\index{Joe|textit}`	Joe, *5*	idem

Les commandes de création de l'index doivent être activées 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 4.5 explique la syntaxe de l'argument *clef* avec plusieurs exemples.

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

`makeindex` *nom de fichier*

Le programme `makeindex` crée un index trié dans le fichier `.idx`. Ensuite, la prochaine fois que le fichier source sera traité, le contenu du fichier `.idx` 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.

4.6 Figures PostScript

Avec les environnements `figure` et `table`, LaTeX fourni 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 comprendre. 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 [5] 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.

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 ». [6].

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 [7]. Cette information est nécessaire car l'insertion de la figure est réalisée par le pilote au moment de l'impression. Cela permet 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

5. Tel que XFig, idraw, CorelDraw!, FreeHand,...
6. `CTAN:/tex-archive/macros/latex/packages/graphics`
7. Le programme le plus répandu est `dvips`.

clefs permettent de modifier la largeur, la hauteur, ou l'angle de rotation de la figure. Le tableau 4.6 présente les clefs les plus importantes.

TAB. 4.6 – *Clefs pour l'extension* graphicx

`width`	définit la largeur de la figure
`height`	définit la hauteur de la figure
`angle`	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].

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 Publishing Company, 1984, ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach and 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] D. P. Carlisle. *Packages in the 'graphics' bundle.* Distribué avec les extensions « graphics » dans `grfguide.tex`.

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

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

Index