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

ou LaTeX2e en 159 minutes

par Tobias Oetiker

Hubert Partl, Irene Hyna et Elisabeth Schlegl

traduit en français par Samuel Colin
et Manuel Pégourié-Gonnard
(à partir de la version 3.21)

ainsi que par Matthieu Herrb
(jusqu'à la version 3.20)

Version 5.01fr-0, 25 Avril 2011

Copyright ©1995-2011 Tobias Oetiker and Contributors. All rights reserved.

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

This document is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fittness for a particular purpose*. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Copyright © 1995-2011 Tobias Oetiker et les contributeurs.

Copyright © 1998-2001 LAAS/CNRS pour la traduction jusqu'à la version 3.20 incluse.

Copyright © 2009-2011 Samuel Colin et Manuel Pégourié-Gonnard pour la traduction à partir de la version 3.21 incluse.

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

Ce document est diffusé en espérant qu'il sera utile, mais *sans aucune garantie*, ni explicite ni implicite, sans même la garantie implicite d'être *commercialisable* ou *adapté à un but spécifique*. Reportez-vous à la licence publique générale de GNU pour plus de détails.

Vous devez avoir reçu une copie de la licence publique générale de GNU en même temps que ce document. Si ce n'est pas le cas, écrivez à la Free Software Fundation, Inc., 675 Mass Ave, Cambridge, MA 02139, États-Unis.

Merci !

Ce document est une traduction en français de « *The not so short introduction to LaTeX2e* » par Tobias Oetiker.
Une grande partie du document sus-cité 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 `<noemail>`
in Graz

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

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

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

1. Voir page viii la liste des sites CTAN.

De nombreuses personnes ont fourni des corrections, des suggestions et du texte pour améliorer ce document. Qu'ils ou elles 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 :

Eric Abrahamsen, Rosemary Bailey, Marc Bevand, Friedemann Brauer, Barbara Beeton, Salvatore Bonaccorso, Jan Busa, Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos, Neil Carter, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke, Chris McCormack, Diego Clavadetscher, Wim van Dam, Benjamin Deschwanden Jan Dittberner, Michael John Downes, Matthias Dreier, David Dureisseix, Eilinger August, Elliot, Rockrush Engch, Hans Ehrbar, Daniel Flipo, David Frey, Hans Fugal, Robert Funnell, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx, Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Andy Goth, Cyril Goutte, Greg Gamble, Frank Fischli, Robert Funnell, Morten Høgholm, Neil Hammond, Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Nils Kanning, Tobias Krewer, Johannes-Maria Kaltenbach, Andrzej Kawalec, Sander de Kievit, Alain Kessi, Christian Kern, Tobias Klauser, Jörg Knappen, Kjetil Kjernsmo, Michael Koundouros, Matt Kraai, Maik Lehradt, Rémi Letot, Flori Lambrechts, Mike Lee, Axel Liljencrantz, Johan Lundberg, Alexander Mai, Hendrik Maryns, Martin Maechler, Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren, Stefan M. Moser, Richard Nagy, Philipp Nagele, Lenimar Nunes de Andrade, I. J. Vera Marún, Manuel Oetiker, Urs Oswald, Marcelo Pasin, Lan Thuy Pham, Martin Pfister, Breno Pietracci, Demerson Andre Polli, Nikos Pothitos, Maksym Polyakov Hubert Partl, John Refling, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Kurt Rosenfeld, Chris Rowley, Axel Kielhorn, Risto Saarelma, Jordi Serra i Solanich, Hanspeter Schmid, Craig Schlenter, Gilles Schintgen, Baron Schwartz, Christopher Sawtell, Miles Spielberg, Matthieu Stigler, Geoffrey Swindale, Laszlo Szathmary, András Salamon, Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Matthew Widmann, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone, et Mikhail Zotov.

Les versions françaises ont bénéficié des contributions des personnes suivantes :

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

Note des traducteurs : nous tenons également à remercier chaleureusement les auteurs de ce document de le rendre publiquement utilisable et d'avoir ainsi rendu possible cette version française. Nous remercions tout aussi chaleureusement Matthieu Herrb pour sa traduction jusqu'à la version 3.20, et dont d'importantes parties se retrouvent encore dans ce document.

Préface

LaTeX [1] est un logiciel de composition typographique particulièrement 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 2ε et devrait se montrer suffisante pour la plupart des applications de LaTeX. Pour une description complète du système LaTeX, reportez-vous à [1, 3].

Cette introduction est composée de six chapitres :

Le chapitre 1 présente la structure élémentaire d'un document LaTeX 2ε. 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 et de son fonctionnement.

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

Le chapitre 3 explique comment produire des formules mathématiques en LaTeX. De nombreux exemples montrent comment utiliser cet atout majeur de LaTeX. À la fin de ce chapitre, vous trouverez des tableaux qui listent tous les symboles mathématiques disponibles sous LaTeX.

Le chapitre 4 traite des index, listes de références bibliographiques et de l'insertion de figures PostScript. Il présente aussi la création de documents PDF avec pdfLaTeX ainsi que quelques autres extensions utiles.

Le chapitre 5 montre comment utiliser LaTeX pour créer des images. Au lieu de dessiner une image à l'aide d'un programme d'infographie donné, la sauvegarder et l'inclure dans LaTeX, vous décrirez l'image et laisserez LaTeX la dessiner pour vous.

Le chapitre 6 contient des informations potentiellement dangereuses. Il vous apprend à modifier la mise en page standard produite par LaTeX et vous permet de transformer les présentations plutôt réussies de LaTeX en quelque chose de laid ou magnifique, selon votre habileté.

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

LaTeX est disponible pour une vaste gamme de systèmes informatiques, des PCs et Macs aux systèmes UNIX [2] et VMS. Dans de nombreuses universités, il est installé sur le réseau informatique, prêt à être utilisé. L'information nécessaire pour y accéder devrait être fournie dans le *Local Guide* [6]. 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.

Si vous avez besoin de récupérer des fichiers relatifs à LaTeX, utilisez les sites CTAN (*Comprehensive TeX Archive Network*). Le site principal est sur `http://www.ctan.org`.

Vous verrez plusieurs références au CTAN au long de ce document, en particulier des pointeurs vers des logiciels ou des documents. Au lieu d'écrire des URL complets, nous avons simplement écrit `CTAN:` suivi du chemin dans l'arborescence du CTAN.

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

Si vous avez des suggestions de choses à ajouter, supprimer ou modifier 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 `<tobi@oetiker.ch>`
OETIKER+PARTNER AG
Aarweg 15
4600 Olten
Switzerland

Matthieu Herrb `<matthieu.herrb@laas.fr>`
(jusqu'à la version 3.20, cité ici en guise d'hommage)

Manuel Pégourié-Gonnard `<mpg@elzevir.fr>`
Institut de mathématiques de Jussieu, France.
(avait débuté un travail de traduction, désormais repris ici)

Samuel Colin `<scolin@hivernal.org>`
(à partir de la version 3.21fr)

La version courante de ce document est disponible sur `CTAN://info/lshort`.

2. UNIX est une marque déposée de The Open Group.

Contents

List of Figures

List of Tables

Chapter 1

Ce qu'il faut savoir

La première partie de ce chapitre couvre rapidement la philosophie et 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 de formules mathématiques.

Knuth a commencé le développement de TeX en 1977 pour tenter d'exploiter les possibilités du matériel d'impression numérique qui commençait à s'introduire dans le milieu de l'édition de l'époque. En particulier, il souhaitait contrecarrer la baisse de qualité typographique qui touchait ses propres livres et articles. Le TeX que nous connaissons aujourd'hui a été publié en 1982, avec de légères améliorations en 1989 visant à mieux gérer les caractères 8 bits ainsi que plusieurs langages. TeX est connu pour sa très grande stabilité, sa capacité à fonctionner sur toutes sortes d'ordinateurs, et son absence quasi-totale de bogues. Son numéro de version converge vers π et vaut actuellement [1] 3.141592653.

TeX se prononce “Tech”, avec un “ch” comme dans le mot écossais “Loch”. [2] Le « ch » provient de l'alphabet grec où X est la lettre « chi ». TeX est aussi

1. Au moment de la traduction... (NdT)
2. Il est à noter que l'orthographe particulière de TeX tend à laisser la prononciation suivre la façon dont les lettres qui le composent sont prononcées dans le pays où il est utilisé. Les allemands, plutôt que d'utiliser le “ch” de “Ach”, préfèrent celui de “Pech”, ce qui donnerait comme prononciation en français “tèche”. À propos de ce point, Knuth a écrit dans le Wikipedia allemand : *Je ne m'offusque pas que les gens prononcent TeX de la manière qu'ils préfèrent ... et en allemand, nombreux sont ceux qui utilisent le “ch” doux car le X suit la voyelle “e”, pas comme le “ch” qui suit le “a”. En Russie,*

la première syllabe du mot grec τεχνολογία (technologie). En alphabet phonétique cela donne [**tex**]... Dans un environnement ASCII, TEX devient TeX.

1.1.2 LATEX

LATEX permet à un auteur de mettre en page et imprimer son travail avec la meilleure qualité typographique en utilisant un format professionnel prédéfini. LATEX a été écrit par Leslie Lamport [1]. Il utilise TEX comme outil de mise en page. Actuellement LATEX est maintenu par Frank Mittelbach [3].

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

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

1.2 Les bases

1.2.1 Auteur, éditeur et typographe

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

Un éditeur humain essaye de comprendre ce que l'auteur avait en tête en écrivant le manuscrit. Il décide de la présentation des en-têtes de chapitres, citation, exemples, formules, etc. 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 de l'information sur la structure logique de son document. Cette information est insérée dans le texte sous la forme de “commandes LATEX”.

Cette approche est totalement différente de l'approche WYSIWYG [4] utilisée par les traitements de texte modernes tels que *Microsoft Word* ou *LibreOffice*. Avec ces programmes, l'auteur définit la mise en page du document de manière interactive pendant la saisie du texte. 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 à l'impression.

“tex” est un mot commun qui se prononce “tyekh”. Je crois cependant que la meilleure prononciation est la grecque, où l'on a le “ch” dur de “ach” et “Loch”.

3. Et l'équipe du projet LATEX 3. (NdT)
4. What you see is what you get – Ce que vous voyez est ce qui sera imprimé.

1.2.2 Choix de la mise en page

La typographie est un métier (un art ?). Les auteurs inexpérimentés font souvent de graves erreurs en considérant que la mise en page est avant tout une question d'esthétique : "si un document est beau, il est bien conçu". Mais un document doit être lu et non accroché dans une galerie d'art. La lisibilité et la compréhensibilité sont bien plus importantes que l'apparence. 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, mais cependant assez pour remplir la page de manière harmonieuse.

Avec un logiciel WYSIWYG, l'auteur produit généralement des documents esthétiquement plaisants (quoi que...) mais très peu ou mal structurés. LaTeX empêche de telles erreurs de formatage en forçant l'auteur à décrire la structure logique de son document et en choisissant lui-même la mise en page la plus [5] appropriée.

1.2.3 Avantages et inconvénients

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

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

- mise en page professionnelle qui donne aux documents l'air de sortir de l'atelier d'un imprimeur ;
- la composition des formules mathématiques se fait de manière pratique ;
- il suffit de connaître quelques commandes de base pour décrire la structure logique du document. Il n'est pas nécessaire de se préoccuper de la mise en page ;
- des structures complexes telles que des notes de bas de page, des renvois, la table des matières ou les références bibliographiques sont faciles à produire ;
- pour la plupart des tâches de typographie qui ne sont pas directement gérées par LaTeX, il existe des extensions gratuites : par exemple pour inclure des figures PostScript ou pour formater une bibliographie selon un standard précis. La majorité de ces extensions sont décrites dans *le LaTeX Companion* [3] et dans *LaTeX, apprentissage, guide et référence* [5] (en français) ;

5. L'un des traducteurs n'est pas aussi optimiste et pense que LaTeX choisit seulement *une* mise en pages appropriée, ce qui n'est déjà pas mal. (NdT)

- LaTeX encourage les auteurs à écrire des documents bien structurés, parce que c'est ainsi qu'il fonctionne (en décrivant la structure) ;
- TeX, l'outil de formatage de LaTeX 2_ε, est réellement portable et gratuit. Le système peut ainsi fonctionner sur quasiment tout type de machine existante.

LaTeX a également quelques inconvénients ; il est difficile pour moi d'en trouver, mais d'autres vous en citeront des centaines :
- LaTeX ne fonctionne pas bien pour ceux qui ont vendu leur âme ;
- bien que quelques paramètres des mises en page pré-définies puissent être personnalisés, la mise au point d'une présentation entièrement nouvelle est difficile et demande beaucoup de temps [6] ;
- écrire des documents mal organisés et mal structurés est très difficile.
- il est possible que votre hamster, malgré des débuts encourageants, ne parvienne jamais à bien comprendre la notion de balisage logique.

1.3 Fichiers source LaTeX

L'entrée de LaTeX est un fichier texte brut. Sous Unix/Linux les fichiers textes sont communs. Sous Windows, d'aucuns utiliseraient Notepad pour créer un fichier texte. Ce fichier 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* [7]. Si vous utilisez un environnement de développement LaTeX intégré, il contiendra un programme pour créer des fichiers sources LaTeX au format texte.

1.3.1 Espaces

Les caractères d'espacement, tels que les blancs ou les tabulations, sont traités de manière unique comme "espace" par LaTeX. Plusieurs blancs *consécutifs* sont considérés comme *une seule* espace [8]. Les espaces en début de ligne sont en général ignorées et un retour à la ligne unique est traité comme une espace.

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

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

Une ligne vide commence
un nouveau paragraphe.
```

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

6. La rumeur dit que c'est un des points qui devrait être améliorés dans la future version LaTeX3.
7. Couramment élidé en « le source » dans la conversation. (NdT)
8. En langage typographique, *espace* est un mot féminin. (NdT)

1.3.2 Caractères spéciaux

Les symboles suivants sont des caractères réservés qui, soit ont une signification spéciale dans LaTeX, soit ne sont pas disponibles dans toutes les polices. Si vous les saisissez directement dans votre texte, ils ne seront pas imprimés 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 documents en les préfixant par une contre-oblique [9] :

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

$ % ^ & _ { } ~ \

Les autres symboles et bien d'autres encore peuvent être obtenus avec des commandes spéciales à l'intérieur de formules mathématiques ou comme accents. La contre-oblique \ ne peut pas être saisie en ajoutant une contre-oblique devant (\\) : cette séquence est utilisée pour indiquer les coupures de ligne ; utilisez la commande `\textbackslash` [10] pour cela.

1.3.3 Commandes LaTeX

Les commandes LaTeX sont sensibles à la casse des caractères (majuscules ou minuscules) et respectent l'un des deux formats suivants :
- soit elles commencent par une contre-oblique `\` et ont un nom composé uniquement de lettres. Un nom de commande est terminé par une espace, un chiffre ou tout autre caractère qui n'est pas une lettre ;
- soit elles sont composées d'une contre-oblique et d'exactement un caractère autre qu'une lettre.
- plusieurs commandes ont aussi une variante "étoilée"où une étoile (*) est ajoutée au nom de la commande.

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

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

J'ai lu que Knuth classe les gens qui utilisent TeX en TeXniciens et TeXperts.
Nous sommes le April 25, 2011.

9. Aussi nommée "barre oblique inverse", ou parfois *antislash* d'après l'anglais *backslash*. (NdT)
10. Fournie par l'extension textcomp. (NdT)

Certaines commandes prennent un paramètre fourni entre accolades `{ }`. Certaines commandes acceptent des paramètres optionnels qui suivent le nom de la commande entre crochets `[ ]`.

`\`*commande*`[`*paramètre optionnel*`]{`*paramètre*`}`

L'exemple suivant montre quelques commandes LaTeX. Ne vous tracassez pas pour les comprendre, elles seront expliquées plus loin.

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

Penchez-vous !

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

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

1.3.4 Commentaires

Quand LaTeX rencontre un caractère `%` dans le fichier source, il ignore le reste de la ligne en cours, le changement de ligne et tous les espaces au début de la ligne [11] suivante.

C'est utile pour ajouter des notes qui n'apparaîtront pas dans la version imprimée.

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

Ceci est un exemple: anticonstitutionnellement

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

Pour créer des commentaires plus longs, on peut utilier l'environnement `comment` fourni par l'extension verbatim à ajouter en préambule. Vous apprendrez plus loin à utiliser une extension.

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

Voici un autre exemple de commentaires.

Notez cependant que cet environnement n'est pas utilisable à l'intérieur d'autres environnements complexes, tels que certains environnements mathématiques par exemple.

11. Ceci est habituel et n'est pas propre à `%`. (NdT)

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 fonctionnalités au système LaTeX. Pour charger une extension, utilisez la commande :

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

Quand tout le travail de préparation est fait [12], 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 minimal. Un fichier source plus complet est présenté sur la figure 1.2.

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

Figure 1.1: Un fichier LaTeX minimal

1.5 Utilisation typique en ligne de commande

Vous brûlez probablement d'envie d'essayer l'exemple présenté page 7. Voici quelques informations : LaTeX lui-même ne propose pas d'interface graphique ni de jolis boutons à cliquer. Il s'agit simplement d'un programme qui “digère” votre fichier source. Certaines installations de LaTeX ajoutent une interface graphique permettant de cliquer pour lancer la “compilation” de votre document. Sur d'autres systèmes il faudra probablement taper quelques lignes de commande, aussi voici comment convaincre LaTeX de compiler votre fichier d'entrée sur un système à interface textuelle. Notez cependant que ces

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

```
\documentclass[a4paper,11pt]{article}
\usepackage[T1]{fontenc}
\usepackage[english,francais]{babel}
\author{P.~Tar}
\title{Le Minimalisme}
\begin{document}
\maketitle
% insérer la table des matières
\tableofcontents
\section{Quelques mots descriptifs}
Et bien, ici commence mon \oe uvre.
\section{Au revoir, monde}
\ldots{} Et ainsi s'ach\`eve mon ouvrage.
\end{document}
```

Figure 1.2: Exemple d'un article de revue plus réaliste. Les commandes que vous voyez dans cet exemple vous seront expliquées plus tard dans cette introduction.

explications supposent que LaTeX soit déjà installé et fonctionnel sur votre ordinateur.[13]

1. Créez/éditez votre fichier source LaTeX. Il s'agit d'un fichier texte pur. Sur les systèmes Unix, tous les éditeurs créent ce type de fichier. Sous Windows, assurez-vous que le fichier est sauvegardé en texte seul (*plain text*). Choisissez pour votre fichier un nom avec le suffixe `.tex`.

2. Ouvrez un terminal ou une ligne de commande, déplacez-vous via la commande `cd` dans le répertoire où se trouve votre fichier et exécutez LaTeX sur celui-ci. Si tout se passe bien, vous obtiendrez un nouveau fichier avec le suffixe `.dvi`. Il peut être nécessaire d'exécuter LaTeX plusieurs fois afin que la table des matières et les références croisées soient à jour. S'il y a une erreur dans votre fichier, LaTeX vous le signalera et s'arrêtera de le traiter. Appuyez sur la combinaison `Ctrl-D` pour revenir à la ligne de commande.

   ```
   latex document.tex
   ```

3. À présent, vous pouvez visualiser le résultat, le fichier DVI. Il y a plusieurs façons de le faire. Vous pouvez par exemple afficher le fichier

13. C'est le cas de toute bonne déclinaison d'Unix, et ... les Puristes utilisent Unix, donc ... ;-)

à l'écran via

```
xdvi document.dvi
```

Cela ne fonctionne que sous un système Unix avec X11. Si vous utilisez Windows avec la distribution MikTeX, vous pouvez essayer via `yap` (*yet another previewer*[14]).

Vous pouvez également transformer le résultat en PostScript pour impression ou visualisation avec Ghostcript.

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

Avec un peu de chance, votre installation LaTeX contient l'outil `dvipdf`, qui permet de convertir les fichiers `.dvi` directement en `.pdf`.

```
dvipdf document.dvi
```

1.6 La mise en page du document

1.6.1 Classes de documents

La première information dont LaTeX a besoin en traitant un fichier source est le type de document que son auteur est en train de créer. Ce type est spécifié par la commande `\documentclass`.

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

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

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

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

elle informe LaTeX qu'il doit composer ce document comme un *article* avec une taille de caractère de base de *onze points* et qu'il devra produire une mise en page pour une impression *double face* sur du papier au format *A4*[15].

14. Encore un autre prévisualiseur. (NdT)
15. Sans l'option `a4paper`, le format de papier sera américain : 8,5 × 11 pouces, soit 216 × 280 mm.

1.6.2 Extensions

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

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

où *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* [6] peut vous fournir plus d'informations sur les extensions installées sur votre site. *le 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.

Les distributions TeX modernes sont fournies avec un très grand nombre d'extensions préinstallées. Vous pouvez utiliser la commande `texdoc` (sous TeX Live et MacTeX) ou `mthelp` (sous MikTeX) pour accéder à la documentation d'une extension.

Table 1.1: Classes de documents

`article`	pour des articles dans des revues scientifiques, des présentations, des rapports courts, des documentations, des invitations, etc.
`proc`	pour des comptes-rendus de conférence. Cette classe est basée sur la classe `article`.
`minimal`	est aussi réduite que possible. Elle définit uniquement une taille de papier et une police de base. Elle est utilisée principalement à des fins de déboguage.
`report`	pour des rapports plus longs contenant plusieurs chapitres, des petits livres, des thèses, etc.
`book`	pour des vrais livres.
`slides`	pour des transparents. Cette classe utilise de grands caractères sans serif. Voir également la classe Beamer.

Table 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`, … letterpaper définit la taille du papier. La taille 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`.

`onecolumn`, `twocolumn` demandent à LaTeX de formater le texte sur une seule colonne (deux colonnes, respectivement).

`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.

`landscape` change la disposition du mode portrait au mode paysage [a].

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

a. Aussi connus sous les noms “à la française” et “à l'italienne”, respectivement. (NdT)

Table 1.3: Quelques extensions fournies avec LaTeX

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

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

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

ifthen fournit des commandes de la forme
'if... then do... otherwise do....'
Décrite dans `ifthen.pdf`, dans *le LaTeX Companion* [3] et dans *LaTeX, apprentissage, guide et référence* [5].

latexsym permet l'utilisation de la police des symboles LaTeX.
Décrite dans `latexsym.pdf`, dans *le LaTeX Companion* [3] et dans *LaTeX, apprentissage, guide et référence* [5].

makeidx fournit des commandes pour réaliser un index.
Décrite dans ce document, section 4.3, dans *le LaTeX Companion* [3] et dans *LaTeX, apprentissage, guide et référence* [5].

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

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.pdf`.

a. Ce fichier devrait être installé sur votre système et vous devriez être capable de le trouver via la commande `texdoc` ou `mthelp` selon votre distribution. Il en est de même pour les autres fichiers cités dans ce tableau.

1.6.3 Styles de page

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

`\pagestyle{`*style*`}`

Le tableau 1.4 donne la liste des styles prédéfinis.

Table 1.4: Les styles de page de LaTeX

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

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

`\thispagestyle{`*style*`}`

Au chapitre 4, page 91, vous apprendrez comment créer vos propres en-têtes et pieds de pages.

1.7 Les fichiers manipulés

L'utilisateur de LaTeX est amené à cotoyer un grand nombre de fichiers aux suffixes variés et probablement mystérieux. Comme chaque suffixe renseigne sur le type de fichier dont il s'agit, il est utile d'en connaître la signification, voici les suffixes les plus courants. Si vous pensez qu'il en manque, n'hésitez pas à nous le signaler :

`.tex` fichier source TeX ou LaTeX, qui peut être compilé avec les commandes `latex` ou `pdflatex` ;

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

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

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

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

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

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

.pdf votre document compilé, résultat principal d'une compilation par la commande `pdflatex` ;

.dvi signifie *DeVice Independent*, c'est le fichier résultat d'une compilation par la commande `latex`. Il peut être visualisé avec un logiciel approprié, converti en PostScript (par `dvips` par exemple) ou en pdf ;

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

.toc contient le matériel nécessaire à la production de la table des matières, si celle-ci a été demandée. Ce fichier sera lu à la prochaine exécution de LaTeX ;

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

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

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

.idx fichier produit seulement si un index est demandé, il doit être traité par `makeindex` (voir section 4.3 page 90). LaTeX y stocke tous les mots qui iront en index ;

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

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

1.8 Gros documents

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

`\include{`*fichier*`}`

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

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

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

Après avoir rencontré cette commande dans le préambule d'un document, seules les commandes `\include` dont les fichiers sont cités en paramètre de la commande `\includeonly` seront exécutées.

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

`\input{`*fichier*`}`

qui insère simplement le fichier indiqué sans aucun traitement sophistiqué.

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

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

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

Chapter 2

Mise en page

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

2.1 La structure du document et le langage

Par Hanspeter Schmid <hanspi@schmid-werren.ch>

La principale raison d'être d'un texte (à l'exception de certains textes de la littérature contemporaine[1]) est de diffuser des idées, de l'information ou de la connaissance au lecteur. Celui-ci comprendra mieux le texte si ces idées sont bien structurées et il ressentira d'autant mieux cette structure si la typographie utilisée reflète la structure logique et sémantique du contenu.

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

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

En général, on sous-estime complètement l'importance du découpage en paragraphes. Certains ignorent même la signification d'un changement de paragraphe ou bien, notamment avec LaTeX, coupent des paragraphes sans

1. *Différente à tout prix*, traduction du *Different at all cost* du texte original, lui-même une traduction du suisse-allemand UVA (*Um's Verrecken Anders*). (NdT)

le savoir. Cette erreur est particulièrement fréquente lorsque des équations sont présentes au milieu du texte. Étudiez les exemples suivants et essayez de comprendre pourquoi des lignes vides (changements de paragraphe) sont parfois utilisées avant et après l'équation et parfois non. (Si vous ne comprenez pas suffisamment les commandes utilisées, lisez d'abord la suite du chapitre puis revenez à cette section.)

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

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

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

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

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

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

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

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

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

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

2.2 Sauts de ligne et de page

2.2.1 Paragraphes justifiés

Les livres sont souvent 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 [2]. Normalement la première ligne d'un paragraphe est en retrait par rapport à la marge gauche et il n'y a pas d'espace vertical particulière entre deux paragraphes (cf. section 6.3.2).

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

`\\` ou `\newline`

commence une nouvelle ligne sans commencer un nouveau paragraphe.

`\\*`

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

`\newpage`

provoque un saut de page.

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

indiquent les endroits où un saut de ligne ou de page devrait apparaître ou non. L'action de ces commandes peut être paramétrée par l'auteur grâce au paramètre optionnel n qui peut prendre une valeur entre zéro et quatre. En donnant à n une valeur inférieure à quatre, vous laissez à LaTeX la possibilité de ne pas tenir compte de votre commande si cela devait rendre le résultat réellement laid. Ne confondez pas ces commandes "break" avec les commandes "new". Même lorsque vous utilisez une commande "break", LaTeX essaye de

2. Et des règles typographiques propres de chaque pays. (NdT)

justifier le bord droit du texte et d'ajuster la longueur totale de la page, comme expliqué plus loin ; cela peut mener à des trous disgracieux dans votre texte. Si vous voulez réellement commencer une "nouvelle" ligne ou une "nouvelle" page, utilisez la commande "new" correspondante.

LaTeX essaye toujours de trouver les meilleurs endroits pour les retours à la ligne. S'il ne trouve pas de solution pour couper les lignes de manière conforme à ses normes de qualité, il laisse dépasser un bout de ligne sur la marge droite du paragraphe. LaTeX émet alors le message d'erreur "`overfull \hbox`[3]". Cela se produit surtout quand LaTeX ne trouve pas de point de césure dans un mot.[4] 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`[5]" est produit. Souvent, malgré tout, le résultat est acceptable. La commande `\fussy` agit dans l'autre sens, au cas où vous voudriez voir LaTeX revenir à ses exigences normales.

2.2.2 Césure

LaTeX coupe les mots en fin de ligne si nécessaire. Si l'algorithme de césure[6] ne trouve pas l'endroit correct pour couper un mot[7], vous pouvez utiliser les commandes suivantes pour informer TeX de l'exception.

La commande :

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

permet de ne couper les mots cités en argument qu'aux endroits indiqués par "`-`". Cette commande doit être placée dans le préambule et ne doit contenir que des mots composés de lettres ou signes considérés comme normaux par LaTeX. La casse des caractères n'est pas prise en compte. Les informations de césure sont associés au langage actif lors de l'invocation de la commande de césure. Cela signifie que si vous placez une commande de césure dans le préambule, cela influencera la césure de l'anglais[8]. Si vous placez la commande après `\begin{document}` et que vous utilisez une extension comme babel pour le support d'une autre langue, alors les suggestions de césure seront actives pour le langage activé via babel.

3. Débordement horizontal.
4. Bien que LaTeX signale un avertissement lorsque cela arrive et affiche la ligne qui pose problème, celle-ci n'est pas toujours facile à retrouver dans le texte. En utilisant l'option `draft` dans la commande `\documentclass`, ces lignes problématiques seront marquées d'une épaisse marque noire dans la marge de droite.
5. Boîte horizontale pas assez pleine.
6. *Hyphenation* en anglais.
7. Ce qui est normalement plutôt rare. Si vous observez de nombreuses erreurs de césure, c'est probablement un problème de spécification de la langue du document ou du codage de sortie. Voir le paragraphe sur le support multilingue, page 26.
8. Par défaut les documents sont supposés être en anglais. (NdT)

L'exemple ci-dessous permet à "anticonstitutionnellement" et "Anticonstitutionnellement", d'être coupés. Mais il empêche toute césure de "FORTRAN", "Fortran" ou "fortran". Ni les caractères spéciaux ni les symboles ne sont autorisés dans cette commande.

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

La commande `\hyphenation{`*liste de mots*`}` a un effet *global* sur toutes les occurrences des mots de la liste. Si l'exception ne concerne qu'une occurrence d'un mot on utilise la commande `\-` qui insère un point de césure potentiel dans un mot. Ces positions deviennent alors les *seuls* points de césure possibles pour cette occurrence du mot. Cette commande est particulièrement utile pour les mots contenant des caractères spéciaux, puisque LaTeX ne réalise pas automatiquement la césure pour ces derniers.

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

I think this is: supercalifragilisticexpiali-
docious

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

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

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

`\mbox{`*texte*`}`

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

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

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

Mon numéro de téléphone va changer. À partir du 18 mai, ce sera le 0561 336 330.

Le paramètre *nom du fichier* de la commande `input` contient le nom du fichier à lire.

`\fbox` est similaire à `\mbox`, à la différence qu'un cadre visible sera en plus dessiné autour du contenu.

2.3 Chaînes prêtes à l'emploi

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

Commande	Résultat	Description
`\today`	April 25, 2011	Date du jour
`\TeX`	TeX	Logo TeX
`\LaTeX`	LaTeX	Logo LaTeX
`\LaTeXe`	LaTeX 2_ε	Sa version actuelle

2.4 Caractères spéciaux et symboles

2.4.1 Guillemets

Pour insérer des guillemets n'utilisez pas le caractère " comme sur une machine à écrire. En typographie, il y a des guillemets ouvrants et fermants spécifiques. En anglais, utilisez deux ` pour les guillemets ouvrants et deux ' pour les guillemets fermants. En français, avec l'option francais de l'extension babel, utilisez `\og` et `\fg` ou bien utilisez directement « et » si vous disposez d'un moyen de saisir ces caractères (voir la section 2.5.2 pour plus de détails).

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

« Appuyez sur la touche `x'. »
```

"Please press the 'x' key."

« Appuyez sur la touche 'x'. »

Je suis conscient que le rendu n'est pas idéal, mais il s'agit effectivement d'un accent grave (`) pour l'ouverture et d'une quote (') (i.e. pas une apostrophe au sens typographique du terme) pour la fermeture, et ce malgré ce que la police choisie semble indiquer.

2.4.2 Tirets

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

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

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

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

2.4.3 Tilde (~)

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

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

http://www.rich.edu/˜bush
http://www.clever.edu/$\sim$demo

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

2.4.4 Barre oblique ou slash (/)

Pour obtenir une barre oblique entre deux mots, il suffit de l'écrire, comme par exemple `lire/écrire`. Cependant LaTeX considère alors cela comme un seul mot et non deux. La césure est désactivée pour ces mots, ce qui peut conduire à des erreurs de débordement de ligne ("overfull hbox"). Pour surmonter ce problème, utilisez `\slash` comme par exemple `lire\slash écrire` qui autorise la césure. Le caractère usuel de barre oblique (/) peut toujours être utilisés pour des fractions ou des unités, par exemple `5 MB/s`.

2.4.5 Symbole degré (°)

L'exemple suivant montre comment obtenir un symbole degré :

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

Il fait $-30\,^{\circ}\mathrm{C}$. Je vais bientôt devenir supra-conducteur.

L'extension textcomp fournit un symbole degré plus adapté, disponible seul avec `\textdegree`, ou accompagné d'un C avec `\textcelsius`.

```
30 \textcelsius{} font
86 \textdegree{}F.
```

30 ℃ font 86 °F.

En français avec l'option `french` de babel, on dispose aussi de la commande `\degres` qui donne un résultat similaire.

2.4.6 Le symbole de l'euro (€)

Écrire sur tout sujet économique de nos jours requiert l'utilisation du symbole de l'euro. De nombreuses polices de caractères contiennent un symbole euro. Après avoir chargé l'extension textcomp dans le préambule

```
\usepackage{textcomp}
```

vous pouvez utiliser la commande

```
\texteuro
```

pour y accéder.

Si votre police ne fournit pas son propre symbole de l'euro ou si vous ne l'aimez pas, il vous reste d'autres possibilités.

Tout d'abord l'extension eurosym qui fournit un symbol officiel de l'euro :

```
\usepackage[official]{eurosym}
```

Si vous préférez un symbole qui se marie bien à votre police, utilisez plutôt `gen` à la place de `official`.

Table 2.1: Un sac plein d'euros

LM+textcomp	`\texteuro`	€	€	€
eurosym	`\euro`	€	€	€
[gen]eurosym	`\euro`	€	€	€

2.4.7 Points de suspension (. . .)

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

```
\dots
```

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

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

2.4.8 Ligatures

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

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

Ces ligatures peuvent être désactivées en insérant un `\mbox{}` entre les lettres en question. Cela peut s'avérer utile pour certains mots composés [9].

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

Not shelfful
but shelfful

2.4.9 Accents et caractères spéciaux

LaTeX permet l'utilisation d'accents et de caractères spéciaux issus de nombreuses langues. Le tableau 2.2 montre tous les accents que l'on peut ajouter à la lettre o. Ils s'appliquent naturellement aux autres lettres.

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

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

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

Table 2.2: Accents et caractères spéciaux

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

9. Il n'existe pas d'exemple en français. (NdT)

2.5 Support multilingue

Pour composer des documents dans des langues autres que l'anglais, il y a plusieurs domaines pour lesquels LaTeX doit s'adapter aux spécificités de chaque langue :

1. Toutes les chaînes de caractères générées automatiquement [10] doivent être traduites. Pour de nombreuses langues, ces adaptations peuvent être réalisées par l'extension babel de Johannes Braams.
2. LaTeX doit connaître les règles de césure de la nouvelle langue. Définir les règles de césure utilisées par LaTeX est une tâche assez complexe, qui impose la construction de formats spécifiques. Votre *Local Guide* [6] devrait vous indiquer quelles sont les langues prises en compte par votre installation de LaTeX et comment configurer les règles de césure.
3. Certaines règles typographiques changent en fonction de la langue ou de la région géographique. Par exemple le français impose une espace avant le caractère deux-points (:).

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

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

après la commande `\documentclass`. La liste des langues intégrées à votre système LaTeX est affichée à l'écran quand vous lancez `latex` et au début du rapport de compilation (fichier `log`). Babel activera automatiquement les règles de césures correctes pour la langue choisie. Si votre format LaTeXne possède pas les règles de césures pour la langue choisie, babel fonctionnera quand même mais désactivera la césure des mots, ce qui peut avoir un impact négatif sur l'aspect de votre document.

Babel définit également pour certaines langues de nouvelles commandes qui simplifient la saisie des caractères spéciaux. L'allemand contient par exemple de nombreux umlauts (äöü). Avec babel, vous pouvez entrer un ö en tapant `"o` au lieu de `\"o`.

Si vous appelez babel avec plusieurs langues,

```
\usepackage[langueA,langueB]{babel}
```

alors la dernière langue dans la liste d'options sera active (c'est-à-dire *langueB*). Vous pouvez utiliser la commande

```
\selectlanguage{langueA}
```

pour changer la langue active au cours de votre document.

10. "Table des matières", "Liste des figures", ...

La plupart des systèmes informatiques modernes vous permettent de saisir directement depuis le clavier les caractères accentués ou les symboles spécifiques de l'alphabet d'une langue. LaTeX 2ε gère la variété des codages d'entrée des différents langages et plateformes grâce à l'extension `inputenc` :

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

En utilisant cette extension, il faut prendre garde au fait que les systèmes informatiques n'utilisent pas tous le même codage des caractères spéciaux. Par exemple, l'umlaut allemand ä est codé 132 sous OS/2, sous les systèmes Unix utilisant ISO-LATIN 1 il est codé 228 et sous le codage cyrillique cp1251 pour Windows cette lettre n'existe même pas; c'est pourquoi vous devez exploiter cette fonctionnalité avec soin.

Pour visualiser correctement votre fichier source sous un autre éditeur, il faut donc sélectionner l'encodage correspondant à celui avec lequel le fichier a été créé. La plupart des éditeurs de texte permettent de choisir l'encodage voulu, mais soyez conscients que certains (notamment sous Windows) ne le permettent pas, et qu'en tout cas cela demande une intervention manuelle.

Par contre, votre fichier sera compilable sans problème sur tout ordinateur, du moment que l'option passée à `inputenc` correspond bien à l'encodage réel du document.

Les encodages suivants peuvent s'avérer utiles en fonction du type de système sous lequel vous travaillez. [11]

Système d'exploitation	encodages Alphabet latin	encodages Alphabet cyrillique
Mac	`applemac`	`macukr`
Unix	`latin1`	`koi8-ru`
Windows	`ansinew`	`cp1251`
DOS, OS/2	`cp850`	`cp866nav`

```
\usepackage[utf8]{inputenc}
```

vous permettra de créer des fichiers d'entrée LaTeX en `utf8`, un codage multioctets où chaque caractère peut nécessiter entre un et quatre octets pour être codé. [12]

Depuis le début de ce siècle la majorité des systèmes d'exploitation sont basés sur Unicode (par exemple, Windows XP, MacOS X). C'est pourquoi il est recommandé d'utiliser `utf8` pour tout nouveau projet. Le codage `utf8`

11. Pour en savoir plus sur les codages d'entrée reconnus par LaTeX, consultez `inputenc.pdf`, `cyinpenc.pdf` et `utf8ienc.pdf`.

12. D'anciennes versions de ce document préconisent la conversion des caractères accentués et spéciaux en séquences du type \' e pour l'échange de documents LaTeX. Cette pratique tend à être de moins en moins nécessaire grâce à Unicode. Il existe cependant des utilitaires pour faire la conversion automatique dans les deux sens : `recode` sous Unix, `Tower of Babel` sur Macintosh, etc. y compris de et vers Unicode. (NdT)

utilisé par inputenc n'est défini que pour les caractères effectivement fournis par les polices utilisées. Si vous avez besoin de plus de caractères (non-latins), vous pouvez regarder la section 4.8 sur XƎLaTeX, une implantation de TeX qui se base sur Unicode.

Pour faciliter la lecture, inputenc est utilisé dans la suite du document pour représenter les caractères accentués dans les exemples.

Le codage des polices de caractères est une autre histoire. Celui-ci définit à quelle position dans une police de TeX se trouve chaque caractère. Plusieurs codages d'entrée sont susceptibles d'être traduits en un codage de fonte donné, ce qui réduit le nombre de dispositions de polices requises. Les codages de polices sont manipulés avec l'extension fontenc :

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

où *codage* est le codage de la police. Il est possible d'en charger plusieurs simultanément.

Le codage de police par défaut de LaTeX est `OT1`, celui de la police TeX Computer Modern originelle. Celui-ci contient seulement les 128 caractères de l'ensemble ASCII sur 7 bits. Au besoin, TeX fabrique des caractères accentués en combinant un caractère normal avec un accent. Bien que le résultat soit visuellement correct, cette approche empêche le système de césure de fonctionner pour les mots contenant des caractères accentués. De plus, certaines lettres latines ne peuvent être fabriquées de cette façon, sans parler des alphabets non-latins, comme le grec ou le cyrillique.

Pour surpasser ces limitations, plusieurs ensembles de polices 8 bits proches de CM furent proposées. Les polices *Extended Cork* (EC) dans le codage `T1` contiennent les lettres et caractères de ponctuation pour la plupart des langues européennes qui utilisent un alphabet latin. L'ensemble de polices LH contient les lettres pour formatter les langages à l'alphabet cyrillique. À cause du nombre imposant de glyphes cyrilliques, ils sont organisés en quatre codages de police : `T2A`, `T2B`, `T2C` et `X2`.[13] Le paquet CB contient des polices dans le codage `LGR` pour la composition de textes en grec.

L'usage de ces polices activera ou améliorera les césures dans les documents non-anglophones. L'autre avantage des polices proches de CM est qu'elles fournissent les polices des familles CM dans tous les poids, formes et tailles optiquement échelonnables de polices.

2.5.1 Support de la langue portugaise

Par Demerson Andre Polli <polli@linux.ime.usp.br>

Pour activer la césure et faire passer tous les textes automatisés en portugais

13. Retrouvez une liste des langages pris en compte par chacun de ces codages dans [15].

Table 2.3: Préambule pour les documents portugais

```
\usepackage[portuguese]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

utilisez la commande :

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

Si vous êtes au Brésil, remplacez le langage par `brazilian`.

Au vu du nombre conséquent d'accents en portugais, vous devriez utiliser

```
\usepackage[latin1]{inputenc}
```

pour les saisir facilement ainsi que

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

pour obtenir des césures correctes.

Voyez le tableau 2.3 qui récapitule ce qu'il y a à ajouter en préambule pour écrire en langue portugaise. Remarquez que nous indiquons ici le codage latin1. Il est préférable d'utiliser utf8 sur un système moderne.

2.5.2 Support de la langue française

Par Daniel Flipo <daniel.flipo@univ-lille1.fr>

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

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

Vous pouvez aussi utiliser l'option `frenchb` qui est un synonyme. L'option `french` a pu être différente à une époque et sur certains systèmes, mais est équivalente aux précédentes sur tous les systèmes depuis 2003. C'est désormais le nom recommandé.

Cette commande active les règles de césure spécifiques du français et adaptent LaTeX à la plupart des règles spécifiques de la typographie française [14] : présentation des listes, insertion automatique de l'espacement avant les signes de ponctuation doubles, etc. Les mots générés automatiquement par LaTeX sont traduits en français et certaines commandes supplémentaires (cf. table 2.4) sont disponibles.

Table 2.4: Commandes de saisie en français

`\og guillemets \fg{}`	“guillemets”
`M\up{me}, D\up{r}`	M$^{\text{me}}$, D$^{\text{r}}$
`1\ier{}, 1\iere{}, 1\ieres{}`	1$^{\text{er}}$, 1$^{\text{re}}$, 1$^{\text{res}}$
`2\ieme{} 4\iemes{}`	2$^{\text{e}}$ 4$^{\text{es}}$
`\No 1, \no 2`	N$^{\text{o}}$ 1, n$^{\text{o}}$ 2
`20~\degres C, 45\degres`	20 °C, 45°
`\bsc{M. Durand}`	M. DURAND
`\nombre{1234,56789}`	1,234.567,89

Vous remarquerez également que la mise en page des listes est changée lors du passage à la langue française. Pour obtenir toutes les informations sur l'option francais de babel et comment modifier son comportement, consultez la partie 29 de `babel.pdf` [14]

Notez également que les versions récentes de francais nécessitent l'extension numprint pour implanter la commande `\nombre`.

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

2.5.3 Support de la langue allemande

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

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

La césure allemande est alors activée, si votre système a été configuré pour cela. Le texte produit automatiquement par LaTeX est traduit en allemand (par ex. “Kapitel”pour un chapitre). De nouvelles commandes (listées dans la table 2.5) permettent la saisie simplifiée des caractères spéciaux même sans utiliser l'extension inputenc. Avec inputenc cette capacité devient un peu vaine mais votre texte est alors un peu enfermé dans un type d'encodage particulier.

Les livres allemands contiennent souvent des marques de citation françaises (“guillemets”). Les typographes allemands les utilisent différemment, cependant. Une citation dans un livre allemand ressemblerait plutôt à »ceci«. En

14. Ou la documentation en français sur la page de son auteur : `http://daniel.flipo.free.fr/frenchb/frenchb2-doc.pdf`. (NdT)

Table 2.5: Caractères spéciaux en allemand

`"a`	"a	`"s`	"s
`"‘`	"‘	`"’`	"’
`"<` ou `\flqq`	"<	`">` ou `\frqq`	">
`\flq`	‹	`\frq`	›
`\dq`	"		

Suisse allemande, les typographes utilisent les “guillemets” comme les français le font.

Un problème majeur découle de l’utilisation de commandes comme `\flq` : si vous utilisez la police OT1 (la police par défaut) les guillemets ressembleront au symbole mathématique “$\ll$”, de quoi causer des maux d’estomac à un typographe. Les polices codées T1, par contre, contiennent les symboles requis. Ainsi, si vous utilisez ce type de marque de citation, assurez-vous d’utiliser le codage T1 (`\usepackage[T1]{fontenc}`).

2.5.4 Support de la langue coréenne [15]

Pour utiliser LaTeX avec la langue coréenne, nous devons résoudre trois problèmes :

1. Nous devons être capables d’éditer des fichiers d’entrée en coréen. Ces fichiers doivent être au format text brut, mais puisque le coréen dispose d’un ensemble de caractères situé en dehors du répertoire US-ASCII, ils apparaîtront étranges dans tout éditeur ASCII usuel. Les codages coréens les plus répandus sont EUC-KR et son extension (avec compatibilité ascendante) utilisée dans la version coréenne de Windows, CP949/Windows-949/UHC. Dans ces codages, chaque caractère US-ASCII représente son caractère usuel, comme dans d’autres codages compatibles avec US-ASCII (ISO-8859-*x*, EUC-JP, Big5 ou Shift_JIS). En revanche les caractères hangul, hanjas (des caractères chinois utilisées en Corée), hangul jamos, hirakanas, katakanas, grecs et cyrilliques, les symboles et les lettres dérivée de KS X 1001 sont représentés par deux octets consécutifs. Le premier possède son propre ensemble MSB. Jusqu’au milieu des années 1990, il fallait beaucoup de temps et d’effort pour installer un environnement capable de gérer la langue coréenne sous un système non localisé. Vous pouvez lire en diagonale la ressource

15. En réponse aux problèmes usuellement rencontrés par les utilisateurs coréens de LaTeX. Cette section fut écrite par Karnes KIM au nom de l’équipe de traduction coréenne de lshort. Elle fut ensuite traduite en anglais par SHIN Jungshik, raccourcie par Tobi Oetiker puis traduite en français par Samuel Colin.

désormais désuette `http://jshin.net/faq` pour vous faire une idée de la difficulté d'utiliser du coréen sous un système non-coréen dans les années 1990. De nos jours les trois systèmes majeurs (Mac OS, Unix, Windows) sont équipés d'un bon support multilingue et de fonctionnalités d'internationalisation, de sorte qu'éditer un fichier texte coréen n'est plus autant un problème, même sur un système non-coréen.

2. TeX et LaTeX ont été conçus à l'origine pour des documents n'ayant pas plus de 256 caractères dans leur alphabet. Pour les faire fonctionner avec des langages ayant considérablement plus de caractères comme le coréen [16] ou le chinois, un mécanisme de "sous-police" a été développé. Il divise une police CJK avec des (dizaines de) milliers de glyphes en un ensemble de sous-polices ayant chacune 256 glyphes. Pour le coréen, il y a déjà 3 extensions couramment utilisées : HLaTeX par UN Koaunghi, hLaTeXp par CHA Jaechoon et l'extension CJK par Werner Lemberg. [17] HLaTeX et hLaTeXp sont spécifiques au coréen et fournissent une localisation coréenne au-delà du support de police. Elles peuvent toutes deux compiler des fichiers textes codés en EUC-KR. HLaTeX peut même compiler des fichiers d'entrée codés en CP949/Windows-949/UHC et UTF-8 si elle est utilisée avec Λ ou Ω.

   L'extension CJK n'est pas spécifique au coréen. Elle peut compiler des fichiers d'entrée en UTF-8 ainsi que différents codages CJK comprenant EUC-KR et CP949/Windows-949/UHC. Elle peut être utilisée pour saisir des documents avec support multilingue (en particulier chinois, japonais et coréen). L'extension CJK n'a pas de localisation coréenne comme celle offerte par HLaTeX et n'est pas livrée avec autant de polices spéciales coréennes qu'HLaTeX.

3. L'idée principale derrière l'utilisation d'un système comme TeX et LaTeX est d'obtenir des documents typographiés d'une manière "esthétiquement" satisfaisante. En conséquence l'élément le plus important en typographie est un ensemble de polices bien conçues. La distribution

16. Le hangul coréen est un script alphabétique avec 14 consonnes de base et 10 voyelles (jamos) de base. À la différence des alphabets latin ou cyrillique, les caractères individuels doivent être arrangés sous forme de groupes rectangulaires de la taille de caractères chinois. Chaque groupe représente une syllabe. Un nombre illimité de syllabes peut être fabriqué avec cet ensemble fini de consonnes et de voyelles. Les standards orthographiques coréens modernes (tant en Corée du Nord qu'en Corée du Sud) imposent cependant des restrictions sur la formation de ces groupes. C'est pourquoi il existe seulement un nombre fini de syllabes syntaxiquement correctes. Le codage coréen définit des points de code individuels pour chacune des syllabes (KS X 1001:1998 et KS X 1002:1992). Ainsi l'hangul, bien qu'alphabétique, est traité comme les systèmes d'écriture chinois et japonais avec des dizaines de milliers de caractères idéographiques/logographiques. l'ISO 10646/Unicode propose les deux façons de représenter l'hangul utilisé pour le coréen *moderne* via l'hangul jamos conjoignant (alphabets: `http://www.unicode.org/charts/PDF/U1100.pdf`) en plus de toutes les syllabes hangul orthographiquement correctes en coréen *moderne* (`http://www.unicode.org/charts/PDF/UAC00.pdf`). L'un des plus éminents défis en typographie coréenne avec LaTeX et autres systèmes similaires est la prise en compte de syllabes du coréen médian — et futur — via la conjonction jamos en Unicode. L'espoir réside dans des moteurs TeX futurs tels que Ω et Λ pour proposer des solutions.

17. Elles peuvent être obtenues depuis `language/korean/HLaTeX/`, `language/korean/CJK/` et `http://knot.kaist.ac.kr/htex/`.

> HLaTeX inclut les polices PostScript UHC de 10 familles différentes et des polices TrueType Munhwabu[18] de 5 différentes familles. L'extension CJK fonctionne avec un ensemble de polices utilisé par des versions antérieures d'HLaTeX et elle peut utiliser la police TrueType Bitstream cyberbit.

Pour utiliser l'extension HLaTeX pour votre document coréen, insérer la déclaration suivante en préambule :

```
\usepackage{hangul}
```

Cette commande active la localisation coréenne. Les textes automatiques (chapitres, sections,. . .) sont traduits en coréen et le format du document est modifié afin de suivre les conventions coréennes. L'extension propose également une "sélection de particules" automatique. En coréen, il y a des paires de particules postfixées équivalentes grammaticalement mais différentes d'apparence. La correction d'une paire dépend du fait que la syllabe qui précéde est une consonne ou une voyelle (la réalité est plus complexe, mais c'est l'idée générale). Les coréens natifs n'ont aucun problème pour choisir la bonne particule, mais il n'est pas possible de déterminer quelle particule utiliser pour les références et pour les textes automatiques qui changeront au fur et à mesure de l'édition du document. Placer les particules appropriées manuellement requiert un douloureux effort à chaque ajout/retrait de référence ou à chaque réordonnancement de parties du document. HLaTeX soulage ses utilisateurs de ce processus ennuyeux et risqué.

Au cas où vous n'avez pas besoin de fonctionnalités de localisation mais souhaitez simplement saisir du texte coréen, insérez plutôt la ligne suivante en préambule :

```
\usepackage{hfont}
```

Pour plus de détails sur la typographie coréenne avec HLaTeX, référez-vous au *HLaTeX Guide*. Consultez également le site web des utilisateurs coréens de TeX (KTUG) sur `http://www.ktug.or.kr/`. Il existe aussi une traduction coréenne de ce manuel.

2.5.5 Support du grec

Par Nikolaos Pothitos <pothitos@di.uoa.gr>

Les commandes à insérer en préambule pour écrire en grec se trouvent dans le tableau 2.6. Ce préambule active la césure et change le texte automatique en grec.[19]

De nouvelles commandes pour une saisie simplifiée du grec deviennent aussi disponibles. Pour passer temporairement en alphabet latin et vice-versa,

18. Ministère coréen de la culture.
19. Si vous ajoutez l'option `utf8x` à `inputenc`, vous pourrez saisir du grec et des caractères unicodes grecs polytoniques.

Table 2.6: Préambule pour les documents grecs

```
\usepackage[english,greek]{babel}
\usepackage[iso-8859-7]{inputenc}
```

vous pouvez utiliser les commandes `\textlatin{`*texte latin*`}` and `\textgreek{`*texte grec*`}`. Elles prennent un argument qui sera formaté avec la police la plus pertinente. Sinon vous pouvez aussi utiliser la commande `\selectlanguage{...}` présentée précédemment. Le tableau 2.7 présente quelques caractères de ponctuation grecs. Vous pouvez utiliser `\euro` pour obtenir le symbole de l'euro.

Table 2.7: Caractères spéciaux grecs

;	·	?	;
((	“	))	”
‘‘	‘	,,	’

2.5.6 Support du cyrillique

Par Maksym Polyakov <polyama@myrealbox.com>

La version 3.7h de babel comprend un support pour les codages `T2*` et pour le formatage des textes bulgares, russes et ukrainiens à base de lettres cyrilliques.

Le support du cyrillique se base sur des mécanismes LaTeX standards en utilisant les extensions fontenc et inputenc. Si cependant vous voulez utiliser du cyrillique en mode mathématique, vous devrez de plus charger mathtext avant fontenc [20] :

```
\usepackage{mathtext}
%\usepackage{amsmath}
\usepackage[T1,T2A]{fontenc}
\usepackage[koi8-ru]{inputenc}
\usepackage[english,bulgarian,russian,ukranian]{babel}
```

En général, babel choisira lui-même le codage de police par défaut : pour les trois langues précitées il s'agit de `T2A`. Les documents ne sont cependant pas limités à un seule codage de police. Pour les documents multilingues avec des alphabets cyrillique et latin, il est raisonnable d'inclure les codages de polices

20. Si vous utilisez les extensions AMS-LaTeX, chargez-les aussi avant fontenc et babel pour éviter des conflits.

latines explicitement. babel prendra en charge le changement vers un codage de fonte approprié lorsqu'un autre langage est sélectionné dans le document.

En plus d'activer les césures, de traduire les textes automatiques et d'activer certaines règles typographiques (comme `\frenchspacing`), babel fournit des commandes additionnelles pour permettre un formatage conforme aux conventions bulgare, russe ou ukrainienne.

Pour ces trois langues, une ponctuation spéciale est fournie. Le tiret cyrillique pour le texte (plus étroit que le tiret latin et entouré d'espaces fines), un tiret pour le dialogue, des marques de citation et des commandes pour faciliter les césures, voir le tableau 2.8.

Table 2.8: Les définitions additionnelles de babel pour le bulgare, le russe et l'ukrainien

`"|`	Désactiver la ligature ici.
`"-`	Un tiret explicite autorisant la césure dans le reste du mot.
`"---`	Tiret long cyrillique pour le texte.
`"--~`	Tiret long cyrillique pour les noms composés.
`"--*`	Tiret long cyrillique pour le dialogue.
`""`	Comme `"-`, mais n'affiche pas le tiret (pour les mots composés avec tiret, p.e. `x-""y` ou d'autres signes comme "disable/enable").
`"~`	Pour une marque de mot composé sans rupture.
`"=`	Pour une marque de mot composé avec rupture, pour autoriser les césures dans les mots le composant.
`",`	Espace fine pour les initiales avec rupture possible pour le nom qui suit.
`"‘`	Pour les marques doubles allemandes de citation (ressemble à „).
`"’`	Pour les marques doubles allemandes de citation, à droite (ressemble à “).
`"<`	Pour les guillemets à gauche (comme «).
`">`	Pour les guillemets à droite (comme »).

Les options babel pour le russe et l'ukrainien définissent les commandes `\Asbuk` et `\asbuk` qui agissent comme `\Alph` et `\alph`[21], mais produisent des majuscules et des minuscules des alphabets russe et ukrainien (en fonction duquel est la langue active du document). L'option bulgare de babel fournit les commandes `\enumBul` et `\enumLat` (`\enumEng`) qui font produire à `\Alph` et `\alph` des lettres des alphabets bulgare ou latin (anglais, resp.). Le comportement par défaut de `\Alph` et `\alph` pour le bulgare est de produire des lettres de l'alphabet bulgare.

21. les commandes pour transformer les compteurs en a, b, c, . . .

2.5.7 Support du mongol

Pour composer du mongol avec LaTeX, vous avec le choix entre deux extensions : babel (multilingue) et MonTeX d'Oliver Corff.

MonTeX prend en compte aussi bien le cyrillique que l'écriture mongole traditionnelle. Pour avoir accès aux commandes de MonTeX, ajoutez

```
\usepackage[langue,encodage]{mls}
```

à votre préambule. Pour obtenir les titres et dates en mongol moderne, utilisez l'option xalx comme *langue*. Pour écrire un document complet en mongol traditionnel, choisissez bicig. Cette dernière active la méthode d'entrée de caractères par translittération simplifiée.

Le mode de translittération latine peut être activé ou désactivé avec les commandes suivantes.

```
\SetDocumentEncodingLMC \SetDocumentEncodingNeutral
```

Vous trouverez plus d'information sur MonTeX sur `CTAN://language/mongolian/montex/doc`.

L'écriture cyrillique du mongol est supportée par babel. Les commandes suivantes activent ce support.

```
\usepackage[T2A]{fontenc}
\usepackage[mn]{inputenc}
\usepackage[mongolian]{babel}
```

où `mn` correspond à l'encodage d'entrée `cp1251`. Si vous préférez la modernité, utilisez `utf8` à la place.

2.5.8 L'alternative Unicode

Par Axel Kielhorn `<A.Kielhorn@web.de>`

Unicode est la meilleure approche si vous souhaitez inclure plusieurs langages dans le même document, en particulier si ces langages ne sont pas basés sur un alphabet latin. Il existe deux systèmes TeX capables de prendre de l'Unicode directement en entrée :

XeTeX , développé initialement pour MacOS X mais désormais disponible pour toutes les architectures. Il fut inclus à partir de 2007 dans TeXLive.

LuaTeX , le successeur de pdfTeX. Il fut inclus à partir de 2008 dans TeXLive.

Nous décrivons dans la suite XeLaTeX tel qu'il existe dans TeXLive 2010.

Démarrage rapide

Convertir un fichier LaTeX existant vers un format compatible avec XeLaTeX entraîne la procédure suivante :

1. Sauvegarder le fichier en UTF-8

2. Retirer

```
\usepackage{inputenc}
\usepackage{fontenc}
\usepackage{textcomp}
```

du préambule.

3. Changer

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

en

```
\usepackage{polyglossia}
\setdefaultlanguage[babelshorthands]{langageA}
```

4. Ajouter

```
\usepackage[Ligatures=TeX]{fontspec}
```

au préambule.

L'extension polyglossia[23] est un substitut à babel. Elle prend soin des motifs de césure et des chaînes de textes générées automatiquement. L'option `babelshorthands` active la compatibilité avec babel en ce qui concerne les raccourcis pour l'allemand et le catalan.

L'extension fontspec[25] gère le chargement des polices pour XƎLaTeX et LuaTeX. La police par défaut est Latin Modern Roman. Il est à noter ici un fait peu connu, à savoir que certaines commandes TeX sont des ligatures définies dans les polices Computer Modern. Si vous souhaitez les utiliser avec une police non-TeX, vous devez les émuler. L'option

C'est du $\Gamma\rho\epsilon\epsilon\kappa$ pour moi

+

Jusqu'à maintenant l'utilisation d'un système TeX Unicode n'apportait aucun avantage. Cet état de fait change dès que l'on quitte l'alphabet latin et que l'on utilise un langage plus intéressant comme le grec ou le russe. Avec un système basé sur Unicode, pour pouvez simplement [22] entrer les caractères dans votre éditeur et TeX les comprendra.

22. Pour des valeurs suffisamment petites de "simple".

Écrire en plusieurs langues est facile, il suffit de spécifier les langages dans le préambule [23] :

```
\setdefaultlanguage{french}
\setotherlanguage[babelshorthands]{german}
```

Pour écrire un paragraphe en allemand, vous utiliserez l'environnement `german` :

```
Texte en français.
\begin{german}
Deutscher Text.
\end{german}
Encore du texte en français.
```

Si vous avez seulement besoin d'un mot dans une langue autre, utilisez alors la commande `\text`*langage* :

```
Texte en français. \textgerman{Gesundheit} est en fait un mot allemand.
```

Cela peut sembler superflu puisque le seul avantage est alors une césure correcte, mais pour un langage plus exotique le jeu en vaut la chandelle.

Parfois la police principale du document ne contient pas les glyphes requis pour le second langage [24]. La solution est alors de spécifier une police à utiliser pour ce langage. Chaque fois qu'un nouveau langage est activé, polyglossia vérifiera d'abord si une police aura été spécifiée pour ce langage.

```
\newfontfamily\russianfont[Script=Cyrillic,(...)]{(font)}
```

Maintenant vous pouvez écrire :

```
\textrussian{Pravda} est un journal russe.
```

Puisque le document que vous lisez actuellement est codé en Latin1, il m'est impossible de montrer les caractères cyrilliques.

L'extension xgreek[26] offre le support pour écrire autant en ancien grec qu'en grec moderne (monotonique ou polytonique).

Langages à l'écriture de droite à gauche.

Certains langages s'écrivent de gauche à droite, d'autres de droite à gauche (abbrévié en RTL, comme "right to left"en anglais). polyglossia doit faire appel à l'extension bidi[27] [25] pour le support des langages RTL. L'extension bidi

23. les noms des langages sont en anglais, il faut bien un socle commun… (NdT)
24. Latin Modern ne contient pas de caractères cyrilliques.
25. `bidi` ne fonctionne pas avec LuaTEX.

doit être la dernière à être chargée, même après hyperref à qui cette place est usuellement réservée (de plus, comme polyglossia charge bidi, alors polyglossia devra être la dernière extension chargée).

L'extension xepersian[28] propose un support de la langue perse. Elle fournit des commandes LaTeX permettant d'écrire des commandes comme `\section` en perse, ce qui rend cette extension attrayante pour les natifs de cette langue. xepersian est la seule extension disposant du support du kashida avec XeLaTeX. Le développement d'une extension avec support du syriaque selon un algorithme similaire est en cours.

La police IranNastaliq fournie par le SCICT [26] est disponible sur son site web `http://www.scict.ir/Portal/Home/Default.aspx`.

L'extension arabxetex[24] propose le support de plusieurs langages utilisant un alphabet arabe :

- arab (arabe)
- persian (perse)
- urdu
- sindhi (sindhî)
- pashto (ou pachto ou pachtoune)
- ottoman (turc)
- kurdish (kurde)
- kashmiri (Cachemire)
- malay (malais ou jawi)
- uighur (ouïghoure)

Elle offre un mécanisme de mise en correspondance des polices qui permet à XeLaTeX de traiter les entrées à l'aide d'une transcription ASCII ArabTeX.

Les polices avec support de plusieurs langues arabes sont offertes par l'IRMUG [27] sur `http://wiki.irmug.org/index.php/X_Series_2`.

Il n'y a pas d'extension pour l'hébreu parce qu'aucune n'est nécessaire, polyglossia étant suffisante ici. Si cependant vous souhaitez une police convenable pour de l'hébreu en Unicode, SBL Hebrew est gratuite pour un usage non commercial et disponible sur `http://www.sbl-site.org/educational/biblicalfonts.aspx`. Vous pouvez également regarder à la police Ezra SIL distribuée sous la licence SIL Open Font License sur `http://www.sil.org/computing/catalog/show_software.asp?id=76`.

Rappelez-vous simplement d'utiliser les commandes suivantes pour les utiliser :

```
\newfontfamily\hebrewfont[Script=Hebrew]{SBL Hebrew}
\newfontfamily\hebrewfont[Script=Hebrew]{Ezra SIL}
```

26. Supreme Council of +Information and Communication Technology – Conseil suprême pour les technologies de l'information et de la communication. (NdT)
27. Iranian Mac User Group – groupe iranien des utilisateurs de Mac (NdT).

Chinois, japonais et coréen (CJK)

L'extension xeCJK[29] prend soin de la sélection des polices et de la ponctuation pour ces langues.

2.6 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 [28] veulent que l'on insère une espace plus large. Mais si un point suit une lettre majuscule, LaTeX considère qu'il s'agit d'une abréviation et insère alors une espace normale.

Toute exception à ces règles doit être spécifiée par l'auteur du document. Une contre-oblique qui précède une espace génère une espace qui ne sera pas élargie par LaTeX. Un tilde "`~`" produit une espace interdisant tout saut de ligne (dit espace *insécable*). `~` est à utiliser pour éviter les coupures indésirables : on code par exemple `M.~Dupont`. La commande `\@` avant un point indique que celui-ci termine une phrase, même lorsqu'il suit une majuscule.

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?

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

```
\frenchspacing
```

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

2.7 Titres, chapitres et sections

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

Dans la classe de document `article`, les commandes de sectionnement

28. Mais pas celles de la typographie française. C'est pourquoi l'exemple suivant reste en anglais. (NdT)

suivantes sont disponibles :

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

Si vous souhaitez découper votre document sans influencer la numérotation des chapitres ou des sections vous pouvez utiliser la commande :

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

Dans les classes `report` et `book`, une commande de sectionnemnent supérieur est disponible (elle s'intercale entre `\part` et `\section`) :

```
\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 en remplacant le `\title` de chaque article par `\chapter`.

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 [29] en lettres.

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

```
\tableofcontents
```

imprime la table des matières à l'endroit où la commande est invoquée. Un document doit être traité (on dit aussi "compilé") deux fois par LaTeX pour avoir une table des matières correcte. Dans certains cas, un troisième passage est même nécessaire. LaTeX vous indique quand c'est le cas.

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

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

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

spécifique pour la table des matières en argument optionnel avant le titre principal :

```
\chapter[Le LAAS du CNRS]{Le laboratoire
         d'analyse et d'architecture
        des systèmes du Centre national
        de la recherche scientifique}
```

Le titre du document est obtenu par la commande :

`\maketitle`

Les éléments de ce titre sont définis par les commandes :

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

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

Vous trouverez un exemple des commandes citées ci-dessus sur la figure 1.2, page 8.

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

`\frontmatter` doit être la première commande après le début du corps du document (`\begin{document}`), elle introduit le prologue du document. Les numéros de pages sont alors en romain (i, ii, iii, etc.) et les sections non-numérotées, comme si vous utilisiez les variantes étoilées des commandes de sectionnement (p.e. `\chapter*{Preface}`), mais les sections apparaissaient tout de même en table des matières ;

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

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

`\backmatter` se place juste avant la bibliographie et les index. Avec les classes standard de document, cette commmande n'a aucun effet visible.

2.8 Références croisées

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

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

où *marque* est un identificateur choisi par l'utilisateur. LaTeX remplace `\ref` par le numéro de la section, de la sous-section, de la figure, du tableau, ou du théorème où la commande `\label` correspondante a été placée. `\pageref` affichera la page de la commande `\label` correspondante. L'utilisation de références croisées rend nécessaire de compiler deux fois le document : à la première compilation les numéros correspondant aux étiquettes `\label{}` sont inscrits dans le fichier `.aux` et, à la compilation suivante, `\ref{}` et `\pageref{}` peuvent imprimer ces numéros [30].

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

Une référence à cette section ressemble à : "voir section 2.8, page 43."

2.9 Notes de bas de page

La commande :

`\footnote{`*texte*`}`

imprime une note de bas de page en bas de la page en cours. Les notes de bas de page doivent être placées après le mot où la phrase auquel elles se réfèrent [31] Les notes qui se réfèrent à une (partie de) phrase devraient être placées après une virgule ou un point. [32]

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

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

a. Ceci est une note de bas de page.

2.10 Souligner l'importance d'un mot

Dans un manuscrit réalisé sur une machine à écrire, les mots importants sont `valorisés en les` <u>`soulignant`</u> ; on peut obtenir ce résultat en LaTeX

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

31. La typographie française demande une espace fine avant la marque de renvoi à la note. Celle-ci est insérée automatiquement par babel si le français est la langue principale du document, depuis la version 2.0 de frenchb. Auparavant, il fallait utiliser `\AddThinSpaceBeforeFootnotes` dans le préambule. (NdT)

32. Remarquez que les notes de bas de page détournent l'attention du lecteur du corps du document. Après tout, tout le monde lit les notes de bas de page — nous sommes une espèce curieuse, alors pourquoi ne pas plus simplement intégrer tout ce que vous souhaitez dire dans le corps du document ? [33]

33. Un guide ne va pas forcément dans la direction qu'il indique :-).

avec la commande :

`\underline{`*texte*`}`

Dans un ouvrage imprimé, on préfère les *mettre en valeur*[34]. La commande de mise en valeur est :

`\emph{`*texte*`}`

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

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

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

2.11 Environnements

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

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

nom est le nom de l'environnement. Les environnements peuvent être imbriqués, à condition que l'ordre de fermeture soit correct.

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

Les sections suivantes vous présentent (presque) tous les environnements importants.

2.11.1 Listes, énumérations et descriptions

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

Notez que l'option `francais` de l'extension `babel` utilise une présentation des listes simples qui respecte les règles typographiques françaises : utilisation d'un tiret pour les listes simples au loin d'un point épais “ • ”, espaces verticaux réduits.

34. *Emphasize* en anglais.

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

1. Il est possible d'imbriquer les environnements à sa guise :
   – mais cela peut ne pas être très beau,
   – ni facile à suivre.
2. Souvenez-vous :
   Clarté : les faits ne vont pas devenir plus sensés parce qu'ils sont dans une liste,
   Synthèse : une liste peut cependant très bien résumer des faits.

2.11.2 Alignements à gauche, à droite et centrage

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

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

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

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

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

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

Au centre de la terre.

2.11.3 Citations et vers

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

```
Une règle typographique
simple pour la longueur
des lignes :
\begin{quote}
Une ligne ne devrait pas comporter
plus de 66~caractères.
\end{quote}
C’est pourquoi les pages
composées par \LaTeX{} ont des
marges importantes et
les journaux utilisent
souvent plusieurs colonnes.
```

> Une règle typographique simple pour la longueur des lignes :
>
> > Une ligne ne devrait pas comporter plus de 66 caractères.
>
> C’est pourquoi les pages composées par LaTeX ont des marges importantes et les journaux utilisent souvent plusieurs colonnes.

Il existe deux autres environnements comparables : `quotation` et `verse`. L’environnement `quotation` est utile pour des citations plus longues, couvrant plusieurs paragraphes parce qu’il indente ceux-ci. 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 [35].

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

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

> Voici le début d’une fugue de Boris Vian :
>
> > Les gens qui n’ont plus rien
> > à faire
> > Se suivent dans la rue
> > comme
> > Des wagons de chemin de fer.
> >
> > Fer fer fer
> > Fer fer fer
> > Fer quoi faire
> > Fer coiffeur.

2.11.4 Résumé

Lors d’une publication scientifique il est usuel de démarrer celle-ci avec un résumé (*abstract*), censé donner au lecteur une vue d’ensemble de ce qu’il doit attendre du document. LaTeX fournit un environnement `abstract` à cette fin. Normalement `abstract` est utilisé dans les documents de classe `article`.

```
\begin{abstract}
L’abstrait abstract résumé.
\end{abstract}
```

> L’abstrait abstract résumé.

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

2.11.5 Impression *verbatim*

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

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

`\verb+`*texte*`+`

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

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

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

La commande `\dots` ...

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

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

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

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

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

```
comme␣ceci␣:-)␣
```

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

2.11.6 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* de la commande :

`\begin{tabular}[`*position*`]{`*description*`}`

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

Si le texte d'une colonne est trop large pour la page, LaTeX n'insèrera pas automatiquement de saut de ligne. Grâce à `p{largeur}` vous pouvez définir un type spécial de colonne qui fera passer le texte à la ligne comme pour un paragraphe usuel.

L'argument *position* définit la position verticale du tableau par rapport au texte environnant. Utilisez une des lettres `t`, `b` et `c` pour l'aligner en haut (*top*), en bas (*bottom*) ou au centre (*center*) respectivement.

À l'intérieur de l'environnement `tabular`, le caractère `&` est le séparateur de colonnes, `\\` commence une nouvelle ligne et `\hline` insère un filet horizontal. Vous pouvez ajoutez des filets partiels via la commande `\cline{i-j}`, où i et j sont les numéros de colonnes de début et de fin du filet.

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

7C0	hexadécimal
3700	octal
11111000000	binaire
1984	décimal

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

Bienvenue dans ce cadre.
Merci de votre visite.

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

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

sans espace

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

avec espaces

S'il n'y a pas de commande prévue[36] 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 décimale. La commande `\@{,}` dans la description du tableau remplace l'espace normale entre les colonnes par une simple virgule, donnant l'impression d'une seule colonne alignée sur le séparateur décimal. N'oubliez pas de remplacer dans votre tableau le point ou la virgule par un séparateur de colonnes (`&`) ! Un titre peut être placé au-dessus de cette "colonne virtuelle"(en fait, de ces deux colonnes) en utilisant la commande `\multicolumn`.

```
\begin{tabular}{c r @{,} l}
Expression       &
\multicolumn{2}{c}{Valeur} \\
\hline
$\pi$               & 3&1416  \\
$\pi^{\pi}$         & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
$\pi^{-1}$          & 0&3183 \\
\end{tabular}
```

Expression	Valeur
π	3,1416
π^{π}	36,46
$(\pi^{\pi})^{\pi}$	80662,7
π^{-1}	0,3183

Autre exemple d'utilisation de `\multicolumn` :

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

Nom	
Dupont	Jules
Durand	Jacques

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

Parfois les tableaux par défaut de LaTeX donnent une impression d'étroitesse. Si vous voulez leur donner plus d'extension, vous pouvez le faire en modifiant les valeurs de `\arraystretch` et `\tabcolsep` comme dans l'exemple suivant.

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

```
\begin{tabular}{|l|}
\hline
Ces lignes sont\\\hline
à l'étroit\\\hline
\end{tabular}

{\renewcommand{\arraystretch}{1.5}
\renewcommand{\tabcolsep}{0.2cm}
\begin{tabular}{|l|}
\hline
Un tableau\\\hline
moins étroit\\\hline
\end{tabular}}
```

Ces lignes sont
à l'étroit

Un tableau
moins étroit

Si vous voulez seulement augmenter la hauteur d'un ligne dans un tableau, vous pouvez utiliser une réglure de largeur nulle [37]. Donnez à cette réglure la hauteur voulue.

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

Étai…
montant

Les `pt` et `ex` dans l'exemple ci-avant sont des unités TEX. Référez-vous au tableau 6.5 en page 132 pour en savoir plus sur les unités de TEX.

2.12 Objets flottants

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

La solution est de laisser "flotter" les figures et les tableaux qui ne rentrent pas sur la page en cours, vers une page suivante et de compléter la page avec le texte qui suit l'objet "flottant". LATEX fournit deux environnements pour les objets flottants adaptés respectivement aux figures (`figure`) et aux tableaux (`table`). Pour faire le meilleur usage de ces deux environnements, il est important de comprendre comment LATEX traite ces objets flottants de manière interne. Dans le cas contraire ces objets deviendront une cause de frustration intense car LATEX ne les placera jamais à l'endroit où vous souhaitiez les voir.

37. En typographie professionnelle ceci est appelé un montant.

Commençons par regarder les commandes que LaTeX propose pour les objets flottants.Tout objet inclus dans un environnement `figure` ou `table` est traité comme un objet flottant. Les deux environnements flottants ont un paramètre optionnel :

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

appelé *placement*. Ce paramètre permet de dire à LaTeX où vous autorisez l'objet à flotter. Un *placement* est composé d'une chaîne de caractères représentant des *placements possibles*. Reportez-vous au tableau 2.9.

Table 2.9: Placements possibles

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

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

Un tableau flottant peut commencer par exemple par la ligne suivante :

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

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

LaTeX place tous les objets flottants qu'il rencontre 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 [38]. 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 leurs spécifications de placement (sauf `h`, qui n'est plus possible). Tous les nouveaux objets flottants rencontrés dans la suite du texte sont ajoutés à la queue des files. LaTeX respecte scrupuleusement l'ordre d'apparition des objets flottants. C'est pourquoi un objet flottant qui ne peut être placé dans le texte repousse tous les autres à la fin du document.

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

D'où la règle :

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

Essayer d'imposer à LaTeX un emplacement particulier pose souvent problème : si l'objet flottant ne tient pas à l'emplacement demandé, alors il est coincé et bloque le reste des objets flottants. En particulier, l'utilisation de la seule option `[h]` pour un flottant est une idée *à proscrire*, les versions modernes de LaTeX changent d'ailleurs automatiquement l'option `[h]` en `[ht]`.

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

Avec la commande :

`\caption{`*texte de la légende*`}`

vous définissez une légende pour l'objet. Un numéro (incrémenté automatiquement) et le mot "Figure" ou "Table"[39] 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. La commande `\label` doit apparaître *après* la commande `\caption` d'une légende si vous voulez référencer le numéro de cette légende.

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

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

Dans l'exemple ci-dessus[40] 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,

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

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.

Plus loin dans cette introduction à LaTeX, vous apprendrez à inclure des figures PostScript dans vos documents LaTeX 2_ε.

2.13 Protection des commandes “fragiles”

Les arguments de commandes telles que `\section` ou `\caption` etc., peuvent apparaître plusieurs fois dans le document (par exemple dans la table des matières, les hauts de pages. . .), on dit qu'il s'agit d'arguments “mobiles” (*moving arguments*). Certaines commandes, comme `\footnote`, `\phantom` etc., ne produisent pas le résultat escompté quand elles sont exécutées comme argument de commandes de type `\section`. La compilation du document échouera alors. On dit de ces commandes qu'elles sont “fragiles”, ce qui signifie qu'elles ont besoin de la protection (comme nous tous ?). Il faut alors les faire précéder de `\protect` : elles fonctionneront alors correctement même si elles sont utilisées dans des arguments mobiles.

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

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

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

Chapter 3

Formules Mathématiques

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

3.1 L'ensemble AMS-LATEX

Si vous souhaitez saisir des textes mathématiques (avancés), vous devriez utiliser AMS-LATEX. Le paquet AMS-LATEX est une collection d'extensions et de classes pour la saisie mathématique. Nous traiterons ici principalement de l'extension amsmath qui fait partie de ce paquet. AMS-LATEX est produit par l'*American Mathematical Society* [1] et est utilisée extensivement pour la mise en forme de mathématiques. LATEX seul fournit bien quelques fonctionnalités et environnements basiques, mais ils sont relativement limités (voire, la logique s'appliquerait plutôt dans l'autre sens : AMS-LATEX est *illimité* !) et parfois incohérents.

AMS-LATEX fait partie de la distribution de base et est fournie avec toutes les distributions récentes de LATEX [2]. Dans ce chapitre nous supposerons qu'amsmath est chargé en préambule, via `\usepackage`{amsmath}.

3.2 Équations simples

Une formule mathématique peut être mise en forme au fil du texte à l'intérieur d'un paragraphe (*style en-ligne*) ou en interrompant le flot du texte

1. la Société américaine de mathématiques. (NdT)
2. Si la vôtre ne l'a pas, visitez `CTAN://macros/latex/required/amslatex`

pour que la mise en forme soit séparée (*style hors-texte*). Les équations mathématiques *dans* un paragraphe sont entrées entre deux signes `$` :

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

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

```
 \TeX{} se prononce
$\tau\epsilon\chi$\\[5pt]
100~m$^{3}$ d'eau\\[5pt]
J'$\heartsuit$ \LaTeX{}
```

TEX se prononce $\tau\epsilon\chi$

100 m^{3} d'eau

J'$\heartsuit$ LATEX

Il vaut mieux composer les équations ou les formules plus importantes “*hors-texte*”, c'est-à-dire sur une ligne à part, plutôt que d'amocher le paragraphe. Pour cela, on les place entre `\begin{equation}` et `\end{equation}`. Vous pouvez ensuite utiliser `\label` pour marquer un numéro d'équation et vous y référer ailleurs dans le texte via la commande `\eqref`. Si vous voulez plutôt nommer l'équation, utiliser `\tag` à la place.

```
Ajoutez $a$ au carré
et $b$ au carré pour obtenir
$c$ au carré. Ou, en
utilisant une approche plus
matheuse :
 \begin{equation}
   a^2 + b^2 = c^2
 \end{equation}
Einstein a dit
 \begin{equation}
   E = mc^2 \label{intelligent}
 \end{equation}
Il n'a pas dit
 \begin{equation}
  1 + 1 = 3 \tag{idiot}
 \end{equation}
Voici une référence à
\eqref{intelligent}.
```

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

$$a^2 + b^2 = c^2 \tag{3.1}$$

Einstein a dit

$$E = mc^2 \tag{3.2}$$

Il n'a pas dit

$$1 + 1 = 3 \tag{idiot}$$

Voici une référence à (3.2).

Si vous ne voulez pas que LATEX numérote vos équations, utilisez la version étoilée d'`equation`, `equation*` [3], ou mieux encore, entourez votre équation par `\[` et `\]` :

3. Fournie par `amsmath`. Si vous ne l'avez pas chargé, vous pouvez utiliser l'environnement `displaymath` de LATEX. Mais il vaut mieux toujours charger `amsmath`, LATEX seul n'ayant que la version non-étoilée d'`equation`.

```
Ajoutez $a$ au carré
et $b$ au carré pour obtenir
$c$ au carré. Ou, en
utilisant une approche plus
matheuse :
\begin{equation*}
  a^2 + b^2 = c^2
\end{equation*}
ou de façon plus condensée :
\[ a^2 + b^2 = c^2 \]
```

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

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

ou de façon plus condensée :

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

Cependant, si vous avez beaucoup d'équations dans votre document, le source LaTeX risque de devenir moins lisible si vous utilisez `\[` et `\]`, à moins de les laisser seuls sur une ligne. Aussi nous recommandons d'utiliser plutôt `equation` et `equation*` pour lesquels il est plus facile d'activer ou désactiver la numérotation par simple ajout ou retrait des étoiles.

Remarquez que les expressions mathématiques sont formatées différemment selon qu'elles sont composées "en-ligne" ou "hors-texte" :

```
Style en-ligne :
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$.
Style hors-texte :
\begin{equation}
  \lim_{n \to \infty}
  \sum_{k=1}^n \frac{1}{k^2}
  = \frac{\pi^2}{6}
\end{equation}
```

Style en-ligne : $\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$.
Style hors-texte :

$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \tag{3.3}$$

En style en-ligne, utilisez la commande `\smash` sur des (sous-)expressions mathématiques à plusieurs niveaux, que ce soit vers le haut ou vers le bas. Cela incite LaTeX à ne pas prendre en compte la hauteur de ces expressions et permet d'avoir un interligne régulier.

```
Une expression mathématique
$d_{e_{e_p}}$ suivie par une
expression $h^{i^{g^h}}$. Par
opposition à une expression
avec smash \smash{$d_{e_{e_p}}$}
suivie par une expression
\smash{$h^{i^{g^h}}$}.
```

Une expression mathématique $d_{e_{e_p}}$ suivie par une expression $h^{i^{g^h}}$. Par opposition à une expression avec smash $d_{e_{e_p}}$ suivie par une expression $h^{i^{g^h}}$.

3.2.1 Mode mathématique

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

1. la plupart des espaces et des retours à la ligne n'ont aucune signification. Les espaces sont déduites de la logique de la formule ou indiquées à

l'aide de commandes spécifiques telles que `\,`, `\quad` ou`\qquad` (nous reparlerons de cela en section 3.6) ;
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 telle. Pour insérer du texte normal (police et espacement standard) dans une formule, il faut utiliser la commande `\text{...}` (voir également la section 3.7 en page 73).

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

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

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

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

Une mode récente et contestable pousse à utiliser la police *blackboard bold* (gras tableau noir, ainsi appelée car c'est par le doublement des verticales des lettres que l'on simule le gras typographique lorsqu'on ne peut faire autrement) qui est obtenue par la commande `\mathbb` de l'extension amssymb [4] pour désigner les ensembles de nombres entiers, réels, etc. L'exemple précédent devient :

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

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

Référez-vous aux tableaux 3.14 en page 83 et 6.4 en page 127 pour plus de polices mathématiques.

3.3 Éléments d'une formule mathématique

Cette section décrit les commandes les plus importantes du mode mathématique. La plupart des commandes de cette section ne nécessitent pas amsmath, sauf mention explicite, mais chargez-la tout de même.

Les lettres **grecques minuscules** sont saisies de la manière suivante : `\alpha`, `\beta`, `\gamma`, etc. Les lettres **grecques majuscules** sont quant à elles saisies ainsi : `\Gamma`, `\Delta`, etc. [5] sont saisies

Référez-vous au tableau 3.2 en page 78 pour une liste de lettres grecques.

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

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

4. amssymb ne fait pas partie du paquet $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX, mais fait peut-être partie de votre distribution LaTeX. Vérifiez celle-ci ou visitez `CTAN:/fonts/amsfonts/latex/` pour l'obtenir.
5. Il n'y a pas de Alpha, Beta... majuscule dans LaTeX 2_ε parce que c'est le même caractère que le A,B... romain.

Les **indices, exposants et lettres supérieures** sont positionnés en utilisant les caractères _ et ^.

Le tableau 3.3 en page 79 liste de nombreuses relations binaires comme $\subseteq$ et $\perp$.

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

```
$p^3_{ij} \qquad
 m_\text{Knuth}\qquad
\sum_{k=1}^3 k \\[5pt]
 a^x+y \neq a^{x+y}\qquad
 e^{x^2} \neq {e^x}^2$
```

$p^3_{ij} \qquad m_\text{Knuth} \qquad \sum_{k=1}^3 k$

$a^x+y \neq a^{x+y} \qquad e^{x^2} \neq {e^x}^2$

La **racine carrée** est saisie via `\sqrt`. La racine n-ième est produite par la commande `\sqrt[`n`]`. La taille du symbole racine est calculée par LaTeX. Pour obtenir le symbole seul, utilisez `\surd`.

Voyez d'autres sortes de flèches comme $\hookrightarrow$ et $\rightleftharpoons$ dans le tableau 3.6 en page 80.

```
$\sqrt{x} = x^{1/2}
 \quad \sqrt[3]{2}
 \quad \sqrt{x^{2} + \sqrt{y}}
 \quad \surd[x^2 + y^2]$
```

$\sqrt{x} = x^{1/2} \quad \sqrt[3]{2} \quad \sqrt{x^{2} + \sqrt{y}} \quad \surd[x^2 + y^2]$

Bien que les **points** pour indiquer les opérations de multiplication soient normalemet omis, les faire apparaître peut aider à la lecture. Utilisez alors `\cdot` qui imprime un seul point centré. `\cdots` imprime des **points de suspension** centrés (à la manière de points de suspension situés plus haut sur la ligne). `\ldots` imprime des points de suspension normaux. En plus de ces commandes vous avez également `\vdots` pour des points alignés verticalement et `\ddots` qui imprime des points en diagonale. D'autres exemples se trouvent en section 3.5.4.

```
$\Psi = v_1 \cdot v_2
 \cdot \ldots \qquad
 n! = 1 \cdot 2
 \cdots (n-1) \cdot n$
```

$\Psi = v_1 \cdot v_2 \cdot \ldots \qquad n! = 1 \cdot 2 \cdots (n-1) \cdot n$

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

```
$0.\overline{3} =
 \underline{\underline{1/3}}$
```

$0.\overline{3} = \underline{\underline{1/3}}$

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

```
$\underbrace{\overbrace{a+b+c}^6
 \cdot \overbrace{d+e+f}^9}
 _{\text{meaning of life}} = 42$
```

$$\underbrace{\overbrace{a+b+c}^6 \cdot \overbrace{d+e+f}^9}_{\text{meaning of life}} = 42$$

Pour ajouter des accents mathématiques tels que des **flèches** ou des **tildes**, vous pouvez utiliser les commandes du tableau 3.1 p. 78. Les chapeaux et les tildes larges, couvrant plusieurs caractères, sont produits par les commandes `\widetilde` et `\widehat`. Notez bien la différence entre `\hat` et `\widehat` ainsi que le placement de `\bar` pour une variable indicée. La commande `'` produit un prime :

```
$f(x) = x^2 \qquad f'(x)
 = 2x \qquad f''(x) = 2\\[5pt]
 \hat{XY} \quad \widehat{XY}
 \quad \bar{x_0} \quad \bar{x}_0$
```

$f(x) = x^2 \qquad f'(x) = 2x \qquad f''(x) = 2$

$\hat{XY} \quad \widehat{XY} \quad \bar{x_0} \quad \bar{x}_0$

Les **vecteurs** sont en général marqués en ajoutant une flèche au-dessus des noms de variables. Ceci est obtenu par la commande `\vec`. Pour coder le vecteur de A à B, les commandes `\overrightarrow` et `\overleftarrow` sont bien utiles :

```
$\vec{a} \qquad
 \vec{AB} \qquad
 \overrightarrow{AB}$
```

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

Les noms des fonctions doivent souvent être imprimés à l'aide d'une police droite et non en italique comme les variables. C'est pourquoi LaTeX fournit les commandes suivantes pour les fonctions les plus utilisées :

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

```
\begin{equation*}
  \lim_{x \rightarrow 0}
  \frac{\sin x}{x}=1
\end{equation*}
```

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

Les fonctions n'apparaissant pas dans la liste peuvent être déclarées avec `\DeclareMathOperator`. Il y a même une version étoilée pour les fonctions avec des indices ou exposants placés en dessous et en dessus [6]. Cette commande ne

6. Parfois appelés *limites.* (NdT)

peut être utilisée qu'en préambule, aussi les lignes commentées de l'exemple doivent être ajoutées en préambule pour fonctionner.

```
%\DeclareMathOperator{\argh}{argh}
%\DeclareMathOperator*{\Nut}{Nut}
\begin{equation*}
  3\argh = 2\Nut_{x=1}
\end{equation*}
```

$$3 \operatorname{argh} = 2 \operatorname*{Nut}_{x=1}$$

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

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

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

Une **fraction** est produite avec `\frac{`*numérateur*`}{`*dénominateur*`}`. Pour les équations en-ligne, la fraction est réduite pour tenir sur la ligne. Ce style peut aussi s'obtenir hors-texte avec `\tfrac`. L'inverse, c'est-à-dire obtenir des fractions en-ligne selon le style hors-texte, est obtenu avec `\dfrac`. La forme utilisant une barre oblique (1/2) est souvent préférable pour des petits éléments.

```
En style hors-texte :
\begin{equation*}
  3/8 \qquad \frac{3}{8}
  \qquad \tfrac{3}{8}
\end{equation*}
```

En style hors-texte :

$$3/8 \qquad \frac{3}{8} \qquad \tfrac{3}{8}$$

```
En style en-ligne :
$1\frac{1}{2}$~heures \\
ou bien $1\dfrac{1}{2}$~heures
```

En style en-ligne : $1\frac{1}{2}$ heures
ou bien $1\dfrac{1}{2}$ heures

Ici nous utilisons la commande `\partial` utilisée habituellement pour une dérivée partielle :

```
\begin{equation*}
  \sqrt{\frac{x^2}{k+1}}\qquad
  x^\frac{2}{k+1}\qquad
  \frac{\partial^2f}
  {\partial x^2}
\end{equation*}
```

$$\sqrt{\frac{x^2}{k+1}}\qquad x^\frac{2}{k+1}\qquad \frac{\partial^2f}{\partial x^2}$$

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

```
La règle de Pascal est
\begin{equation*}
 \binom{n}{k} =\binom{n-1}{k}
 + \binom{n-1}{k-1}
\end{equation*}
```

La règle de Pascal est

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Il est parfois utile, notamment pour des relations binaires, de pouvoir superposer des symboles. La commande `\stackrel{#1}{#2}` place l'argument `#1` en taille réduite au-dessus de l'argument `#2`, lui-même mis en position normale :

```
\begin{equation*}
 f_n(x) \stackrel{*}{\approx} 1
\end{equation*}
```

$$f_n(x) \stackrel{*}{\approx} 1$$

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

```
\begin{equation*}
\sum_{i=1}^n \qquad
\int_0^{\frac{\pi}{2}} \qquad
\prod_\epsilon
\end{equation*}
```

$$\sum_{i=1}^n \qquad \int_0^{\frac{\pi}{2}} \qquad \prod_\epsilon$$

Pour superposer des indices, l'extension `amsmath` propose la commande `\substack` :

```
\begin{equation*}
\sum^n_{\substack{0<i<n \\
        j\subseteq i}}
   P(i,j) = Q(i,j)
\end{equation*}
```

$$\sum^n_{\substack{0<i<n \\ j\subseteq i}} P(i,j) = Q(i,j)$$

LaTeX fournit toutes sortes de symboles pour les **crochets et autres délimiteurs** (par exemple $[\ \langle\ \|\ \ \updownarrow$). Les parenthèses et les crochets sont obtenus avec les caractères correspondants, les accolades avec `\{`, mais les autres délimiteurs ne sont obtenus que par des commandes spéciales (par exemple `\updownarrow`) :

```
\begin{equation*}
{a,b,c} \neq \{a,b,c\}
\end{equation*}
```

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

Si vous ajoutez `\left` avant un délimiteur ouvrant et `\right` avant le délimiteur fermant correspondant, LaTeX détermine automatiquement la taille appropriée pour ces symboles. Remarquez qu'il est nécessaire de fermer chaque délimiteur ouvrant (`\left`) avec un délimiteur fermant (`\right`). Si vous ne voulez pas de délimiteur fermant, utilisez le délimiteur invisible "`\right.`" :

```
\begin{equation*}
1 + \left(\frac{1}{1-x^{2}}
    \right)^3 \qquad
\left. \ddagger \frac{~}{~}\right)
\end{equation*}
```

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

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

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

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

Pour une liste de tous les délimiteurs disponibles, reportez-vous au tableau 3.8, page 81.

3.4 Formules trop longues : `multline`

Si une équation est trop longue, il faut trouver un moyen de la répartir sur plusieurs lignes. Malheureusement, l'équation a alors tendance à devenir moins lisible. Pour préserver la lisibilité, voici quelques règles sur la façon de découper les équations.

1. En général, une équation devrait toujours être coupée *avant* un signe d'égalité ou un opérateur.
2. Une coupure devant un signe d'égalité est préférable à une coupure devant un opérateur.
3. Une coupure devant un signe "plus"ou moins est préférable à une coupure devant un signe de multiplication.
4. Il faut éviter autant que possible les coupures à tous les autres endroits.

La façon la plus simple pour réaliser un tel découpage est d'utiliser l'environnement `multline` [7].

7. Fourni par amsmath.

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

$$\begin{multline} a+b+c+d+e+f+g+h+i \\ =j+k+l+m+n \tag{3.4}\end{multline}$$

La différence avec l'environnement `equation` est qu'il est possible d'insérer un saut de ligne où l'on veut (voire plusieurs), en utilisant `\\` là où l'équation doit être coupée. Comme pour `equation*`, il existe un environnement `multline*` qui supprime la numérotation.

Souvent le résultat sera meilleur avec l'environnement `IEEEeqnarray` (voir section 3.5). Par exemple, considérons la situation suivante.

```
\begin{equation}
  a = b + c + d + e + f
  + g + h + i + j
  + k + l + m + n + o + p
  \label{eq:formule_trop_longue}
\end{equation}
```

$$a=b+c+d+e+f+g+h+i+j+k+l+m+n+o+p \tag{3.5}$$

Ici, c'est en fait le membre de droite qui est trop long pour tenir sur une ligne. L'usage de `multline` produit alors :

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

$$\begin{multline} a=b+c+d+e+f+g+h+i+j \\ +k+l+m+n+o+p \tag{3.6}\end{multline}$$

C'est nettement mieux que (3.5), mais l'inconvénient est que le signe d'égalité perd son importance naturelle, en principe supérieure à celle du signe "plus"devant k. Une meilleure solution est d'utiliser `IEEEeqnarray`, qui sera présenté en détails en section 3.5.

3.5 Formules multiples

Dans le cas le plus général, considérons une suite d'équations qui ne tiennent pas en entier sur une ligne. Il s'agit d'ajuster l'alignement vertical de façon à obtenir une structure lisible et agréable à l'œil pour l'ensemble.

Avant les conseils pour obtenir ce résultat, commençons par quelques mauvais exemples qui montrent les principaux inconvénients de certaines solutions courantes.

3.5.1 Problèmes des commandes traditionnelles

Pour grouper ensemble plusieurs équations, on peut utiliser l'environnement `align` [8] comme ceci :

```
\begin{align}
 a & = b + c \\
 & = d + e
\end{align}
```

$$a = b + c \tag{3.7}$$
$$= d + e \tag{3.8}$$

Cette approche pose problème dès qu'une ligne est trop longue.

```
\begin{align}
  a & = b + c \\
  & = d + e + f + g + h + i
  + j + k + l \nonumber \\
  & + m + n + o \\
  & = p + q + r + s
\end{align}
```

$$a = b + c \tag{3.9}$$
$$= d + e + f + g + h + i + j + k + l$$
$$+ m + n + o \tag{3.10}$$
$$= p + q + r + s \tag{3.11}$$

Ici, on voudrait que $+\,m$ soit placé exactement sous le d et non sous le signe d'égalité. Bien sûr, il est possible d'ajouter un peu d'espace avec `\hspace{...}`, mais il est difficile d'obtenir un alignement précis ainsi (et ce n'est pas un très bon style de programmation).

L'environnement `eqnarray` offre à première vue une meilleure solution.

```
\begin{eqnarray}
  a & = & b + c \\
  & = & d + e + f + g + h + i
  + j + k + l \nonumber \\
  && +\: m + n + o \\
  & = & p + q + r + s
\end{eqnarray}
```

$$a \quad = \quad b + c \tag{3.12}$$
$$= \quad d + e + f + g + h + i + j + k + l$$
$$+\, m + n + o \tag{3.13}$$
$$= \quad p + q + r + s \tag{3.14}$$

Ça n'est cependant toujours pas une solution optimale puisque les espaces autour du signe d'égalité sont trop grands. En particulier, ce ne sont pas les mêmes qu'avec les environnements `multline` ou `equation` :

```
\begin{eqnarray}
  a & = & a = a
\end{eqnarray}
```

$$a \quad = \quad a = a \tag{3.15}$$

... De plus, la formule et le numéro d'équation peuvent se chevaucher, même quand il y a suffisamment de place à gauche pour pousser la formule :

8. L'environnement `align` peut aussi servir à placer plusieurs groupes d'équations côte à côte. Un autre excellent cas d'utilisation de l'environnement `IEEEeqnarray`, avec un argument comme `{rCl+rCl}`.

```
\begin{eqnarray}
  a & = & b + c
  \\
  & = & d + e + f + g + h^2
  + i^2 + j
  \label{eq:eqnarrayfautif}
\end{eqnarray}
```

$$
\begin{array}{rcll}
a & = & b+c & (3.16) \\
 & = & d+e+f+g+h^2+i^2+j & (3.17)
\end{array}
$$

Alors que l'environnement propose une commande `\lefteqn` qui peut être utilisée quand le membre de gauche est trop long.

```
\begin{eqnarray}
  \lefteqn{a + b + c + d
    + e + f + g + h}\nonumber\\
  & = & i + j + k + l + m
  \\
  & = & n + o + p + q + r + s
\end{eqnarray}
```

$$
\begin{array}{rcll}
\multicolumn{3}{l}{a+b+c+d+e+f+g+h} & \\
 & = & i+j+k+l+m & (3.18) \\
 & = & n+o+p+q+r+s & (3.19)
\end{array}
$$

Ça n'est toujours pas optimal étant donné que le membre de droite est trop court et l'ensemble n'est pas centré correctement :

```
\begin{eqnarray}
  \lefteqn{a + b + c + d
    + e + f + g + h}
  \nonumber \\
  & = & i + j
\end{eqnarray}
```

$$
\begin{array}{rcll}
\multicolumn{3}{l}{a+b+c+d+e+f+g+h} & \\
 & = & i+j & (3.20)
\end{array}
$$

Après avoir suffisamment dénigré la compétition, nous pouvons nous diriger vers le glorieux...

3.5.2 L'environnement IEEEeqnarray

L'environnement `IEEEeqnarray` est très puissant et offre beaucoup d'options. Nous n'en couvrirons ici que les fonctionnalités de base. Pour plus d'information, nous renvoyons à son manuel [9].

Pour commencer, il faut charger l'extension [10] IEEEtrantools afin de pouvoir utiliser `IEEEeqnarray`. Pour cela, ajoutez la ligne suivante à votre préambule.

```
\usepackage[retainorgcmds]{IEEEtrantools}
```

Ce qui fait la force d'`IEEEeqnarray`, c'est la capacité de spécifier le nombre de *colonnes* dans le tableau d'équation. Le plus souvent, cette spécification sera `{rCl}`, c'est-à-dire trois colonnes dont la première est alignée à droite, la deuxième centrée et avec un peu plus d'espace autour d'elle (c'est le sens du `C` majuscule au lieu de `c` minuscule), et la troisième alignée à gauche.

9. Le manuel officiel s'appelle `IEEEtran_HOWTO.pdf` ; la partie sur `IEEEeqnarray` se trouve dans l'appendice F.

10. L'extension IEEEtrantools peut ne pas être présente dans votre installation, vous pourrez alors la trouver sur le CTAN.

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k \nonumber\\
  && +\: l + m + n + o
  \\
  & = & p + q + r + s
\end{IEEEeqnarray}
```

$$
\begin{aligned}
a &= b+c && (3.21)\\
&= d+e+f+g+h+i+j+k &&\\
&\quad +l+m+n+o && (3.22)\\
&= p+q+r+s && (3.23)
\end{aligned}
$$

On peut choisir n'importe quel nombre de colonnes. Par exemple, `{c}` donnera une seule colonne dans laquelle toutes les lignes seront centrées, ou bien `{rCll}` donnera une quatrième colonne alignée à gauche, par exemple pour des commentaires. Mieux, en plus de `l`, `c`, `r`, `L`, `C`, `R` pour les entrées en mode mathématique, on dispose aussi de `s`, `t`, `u` pour des colonnes en mode texte, respectivement alignées à gauche, centrées ou alignées à droite. Par ailleurs, on peut ajouter de l'espace entre deux colonnes avec[11] "`.`", "`/`"et "`?`"en ordre croissant. Remarquons les espaces autour des signes d'égalité, par rapport au cas d'utilisation de l'environnement `eqnarray`.

3.5.3 Usages courants

Nous décrivons ici comment utiliser `IEEEeqnarray` pour résoudre les problèmes les plus courants.

Si une ligne et le numéro d'équation se superposent comme dans (3.17), la commande

```
\IEEEeqnarraynumspace
```

permet de résoudre le problème : il suffit de l'ajouter à la fin de la ligne en question, et l'ensemble de la formule est décalée vers la gauche de la longueur nécessaire (le décalage dépend de la taille du numéro d'équation). Par exemple, partant de :

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k
  \\
  & = & l + m + n
\end{IEEEeqnarray}
```

$$
\begin{aligned}
a &= b+c && (3.24)\\
&= d+e+f+g+h+i+j+k && (3.25)\\
&= l+m+n && (3.26)
\end{aligned}
$$

on obtient :

11. Pour plus de types d'espacement, nous renvoyons à la section 3.8.1 et au manuel officiel.

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k
  \IEEEeqnarraynumspace\\
  & = & l + m + n.
\end{IEEEeqnarray}
```

$$\begin{aligned} a &= b + c && (3.27)\\ &= d+e+f+g+h+i+j+k && (3.28)\\ &= l+m+n. && (3.29)\end{aligned}$$

Si le membre de gauche est trop long, `IEEEeqnarray` propose la commande `\IEEEeqnarraymulticol`, qui fonctionne dans tous les cas, comme remplacement de `\lefteqn`.

```
\begin{IEEEeqnarray}{rCl}
  \IEEEeqnarraymulticol{3}{l}{
    a + b + c + d + e + f
    + g + h
  }\nonumber\\ \quad
  & = & i + j
  \\
  & = & k + l + m
\end{IEEEeqnarray}
```

$$\begin{aligned} &a+b+c+d+e+f+g+h\\ &\quad = i+j && (3.30)\\ &\quad = k+l+m && (3.31)\end{aligned}$$

Son usage est identique à celui de la commande `\multicolumns` dans l'environnement `tabular` : le premier argument `3` dit qu'il faut combiner trois colonnes pour n'en former qu'une, et le deuxième argument `l` dit que son contenu sera aligné à gauche.

Remarquons qu'en insérant `\quad` dans l'exemple précédent, on peut facilement adapter l'alignement vertical[12] du signe égal, par exemple :

```
\begin{IEEEeqnarray}{rCl}
  \IEEEeqnarraymulticol{3}{l}{
    a + b + c + d + e + f
    + g + h
  }\nonumber\\ \qquad\qquad
  & = & i + j
  \\
  & = & k + l + m
\end{IEEEeqnarray}
```

$$\begin{aligned} &a+b+c+d+e+f+g+h\\ &\qquad\qquad = i+j && (3.32)\\ &\qquad\qquad = k+l+m && (3.33)\end{aligned}$$

Si une formule est découpée en deux ou plusieurs lignes, LaTeX croit que le signe + ou − au début d'une ligne est un opérateur unaire (comme dans -2) et non binaire (comme dans $2-2$). C'est pourquoi il faut ajouter un peu d'espace `\:` entre ce signe et le terme qui suit. Par exemple, au lieu de

12. Nous pensons qu'un cadratin est la distance qui rend le mieux dans la plupart des cas.

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k \nonumber\\
  && + l + m + n + o
  \\
  & = & p + q + r + s
\end{IEEEeqnarray}
```

$$\begin{aligned} a &= b + c && (3.34)\\ &= d + e + f + g + h + i + j + k \\ &\quad {+l} + m + n + o && (3.35)\\ &= p + q + r + s && (3.36)\end{aligned}$$

il vaut mieux écrire

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \\
  & = & d + e + f + g + h
  + i + j + k \nonumber\\
  && +\: l + m + n + o
  \\
  & = & p + q + r + s
\end{IEEEeqnarray}
```

$$\begin{aligned} a &= b + c && (3.37)\\ &= d + e + f + g + h + i + j + k \\ &\quad +\: l + m + n + o && (3.38)\\ &= p + q + r + s && (3.39)\end{aligned}$$

Observez l'espace entre $+$ et l dans les deux cas !

Parfois, un espacement adéquat suffit, comme par exemple devant
- un nom d'opérateur comme `\log`, `\sin`, `\det`, `\max`, etc. ;
- une intégrale `\int` ou une somme `\sum` ;
- une parenthèse dont la taille s'adapte en utilisant `\left` ou `\right` (par opposition à une parenthèse simple, ou dont la taille est fixée avec par exemple `\big`) ;

un signe $+$ ou $-$ ne peut pas être unaire et doit être un opérateur binaire. Dans ces cas, LaTeX ajuste correctement l'espace après le signe, et il ne faut pas en rajouter.

Pour supprimer la numérotation sur une ligne particulière, on dispose de la commande `\nonumber` (ou bien `\IEEEnonumber`). Si un label `\label{...}` est défini sur cette ligne, il est alors transmis à la prochaine ligne dont la numérotation n'est pas supprimée. Placez plutôt les labels juste avant le `\\` à la fin de la ligne concernée ou avant la fin de la formule. En plus d'améliorer la lisibilité du code source, ceci évite des erreurs de compilation dans le cas où une commande `\IEEEmulticol` se trouverai après la définition du label.

Il existe également une version étoilée de l'environnement, qui supprime totalement la numérotation. Dans ce cas, on peut ajouter un numéro d'équation sur une ligne particulière avec `\IEEEyesnumber`.

```
\begin{IEEEeqnarray*}{rCl}
  a & = & b + c \\
  & = & d + e \IEEEyesnumber\\
  & = & f + g
\end{IEEEeqnarray*}
```

$$\begin{aligned} a &= b + c \\ &= d + e && (3.40)\\ &= f + g\end{aligned}$$

On peut également obtenir des numéros de sous-équation avec la commande `\IEEEyessubnumber`.

```
\begin{IEEEeqnarray}{rCl}
  a & = & b + c
  \IEEEyessubnumber\\
  & = & d + e
  \nonumber\\
  & = & f + g
  \IEEEyessubnumber
\end{IEEEeqnarray}
```

$$
\begin{aligned}
a &= b + c && \text{(3.41a)}\\
&= d + e &&\\
&= f + g && \text{(3.41b)}
\end{aligned}
$$

3.5.4 Tableaux et matrices

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

```
\begin{equation*}
  \mathbf{X} =
  \left(
    \begin{array}{ccc}
      x_1 & x_2 & \ldots \\
      x_3 & x_4 & \ldots \\
      \vdots & \vdots & \ddots
    \end{array}
  \right)
\end{equation*}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_1 & x_2 & \ldots \\ x_3 & x_4 & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

L'environnement `array` peut également être utilisé pour imprimer des fonctions définies par morceaux en utilisant "`.`" comme délimiteur (invisible) de droite :

```
\begin{equation*}
  |x| =
  \left\{
    \begin{array}{rl}
      -x & \text{si } x < 0,\\
      0 & \text{si } x = 0,\\
      x & \text{si } x > 0.
    \end{array}
  \right.
\end{equation*}
```

$$|x| = \left\{ \begin{array}{rl} -x & \text{si } x < 0,\\ 0 & \text{si } x = 0,\\ x & \text{si } x > 0. \end{array} \right.$$

L'environnement `cases` d'`amsmath` offre une syntaxe plus simple.

```
\begin{equation*}
  |x| =
  \begin{cases}
    -x & \text{si } x < 0,\\
    0 & \text{si } x = 0,\\
    x & \text{si } x > 0.
  \end{cases}
\end{equation*}
```

$$|x| = \begin{cases} -x & \text{si } x < 0, \\ 0 & \text{si } x = 0, \\ x & \text{si } x > 0. \end{cases}$$

L'environnement `array` peut servir à mettre en page des matrices, mais `amsmath` fournit une meilleure solution avec l'environnement `matrix` et ses variantes. Elles sont au nombre de six (avec des délimiteurs différents) : `matrix` (aucun délimiteur), `pmatrix` (, `bmatrix` [, `Bmatrix` {, `vmatrix` | et `Vmatrix` ‖. Vous n'avez pas à spécifier le nombre de colonnes comme avec `array`. Leur nombre maximal est de 10 par défaut mais il est modifiable (bien que ce ne soit pas fréquent d'avoir besoin de 10 colonnes ou plus !) :

```
\begin{equation*}
  \begin{matrix}
    1 & 2 \\
    3 & 4
  \end{matrix} \qquad
  \begin{bmatrix}
    p_{11} & p_{12} & \ldots
    & p_{1n} \\
    p_{21} & p_{22} & \ldots
    & p_{2n} \\
    \vdots & \vdots & \ddots
    & \vdots \\
    p_{m1} & p_{m2} & \ldots
    & p_{mn}
  \end{bmatrix}
\end{equation*}
```

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \qquad \begin{bmatrix} p_{11} & p_{12} & \ldots & p_{1n} \\ p_{21} & p_{22} & \ldots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \ldots & p_{mn} \end{bmatrix}$$

3.6 Espacement en mode mathématique

Si l'espacement choisi par LaTeX dans une formule n'est pas satisfaisant, il peut être ajusté en insérant des commandes d'espacement. Les plus importantes sont : `\,` pour une espace fine ($\frac{3}{18}$ quad, ␣), `\:` pour une espace moyenne ($\frac{4}{18}$ quad, ␣) et `\;` pour une espace grande ($\frac{5}{18}$ quad, ␣). L'espace échappée `\␣` crée une espace moyenne similaire à l'espace entre mots. and `\quad` (␣) et `\qquad` (␣) produisent des espaces plus larges. La largeur d'un `\quad` [13] correspond approximativement à la largeur du caractère "M" dans la police courante. La commande `\!` produit une espace fine négative de $-\frac{3}{18}$ quad (−␣).

13. Appelé *espace cadratin* en français. (NdT)

```
\begin{equation*}
  \int_1^2 \ln x \mathrm{d}x
  \qquad
  \int_1^2 \ln x \,\mathrm{d}x
\end{equation*}
```

$$\int_1^2 \ln x \mathrm{d}x \qquad \int_1^2 \ln x \,\mathrm{d}x$$

Remarquez que "d" est imprimée en police romaine. Dans l'exemple suivant, nous définissons une commande `\ud` qui produit " d" (remarquez l'espace ␣ avant le d), de manière à ne pas avoir à le saisir à chaque fois. La commande `\newcommand` est placée en préambule.

```
\newcommand{\ud}{\,\mathrm{d}}

\begin{equation*}
 \int_a^b f(x)\ud x
\end{equation*}
```

$$\int_a^b f(x)\,\mathrm{d}x$$

Lorsque vous utilisez des intégrales multiples, vous constatez que l'espace entre celles-ci est trop grand. Vous pouvez certes utiliser `\!`, mais $\mathcal{AMS}$-LaTeX propose un ensemble de commandes pour réaliser cet ajustement : `\iint`, `\iiint`, `\iiiint` et `\idotsint`.

```
\newcommand{\ud}{\,\mathrm{d}}

\begin{IEEEeqnarray*}{c}
  \int\int f(x)g(y)
                  \ud x \ud y \\
  \int\!\!\!\int
        f(x)g(y)  \ud x \ud y \\
  \iint f(x)g(y)  \ud x \ud y
\end{IEEEeqnarray*}
```

$$\begin{gathered}
\int\int f(x)g(y)\,\mathrm{d}x\,\mathrm{d}y \\
\int\!\!\!\int f(x)g(y)\,\mathrm{d}x\,\mathrm{d}y \\
\iint f(x)g(y)\,\mathrm{d}x\,\mathrm{d}y
\end{gathered}$$

Reportez-vous au document `testmath.tex` distribué avec $\mathcal{AMS}$-LaTeX, au chapitre 8 de *le LaTeX Companion* [3] ou au chapitre 9 de *LaTeX, apprentissage, guide et référence* [5] pour plus de détails.

3.6.1 Fantômes...

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

```
\begin{equation*}
{}^{14}_{6}\text{C}
\qquad \text{à comparer à} \qquad
{}^{14}_{\phantom{1}6}\text{C}
\end{equation*}
```

$${}^{14}_{6}\text{C} \qquad \text{à comparer à} \qquad {}^{14}_{6}\text{C}$$

Si vous souhaitez mettre en forme des isotopes comme dans l'exemple ci-avant, l'extension mhchem dédiée aux formules chimiques peut vous y aider.

3.7 Manipuler les polices mathématiques

Plusieurs polices mathématiques sont listées au tableau 3.14 page 83.

```
$\Re \qquad
 \mathcal{R} \qquad
 \mathfrak{R} \qquad
 \mathbb{R} \qquad $
```

$\Re \qquad \mathcal{R} \qquad \mathfrak{R} \qquad \mathbb{R} \qquad$

Les deux dernières nécessitent amssymb ou amsfonts.

Parfois, il peut être nécessaire d'indiquer à LaTeX la taille de fonte désirée. En mode mathématique, cette taille est ajustable avec les quatre commandes :

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

Si $\sum$ est placé dans une fraction, il sera imprimé dans le style "en-ligne"à moins d'indiquer à LaTeX le contraire :

```
\begin{equation*}
 P = \frac{\displaystyle{
   \sum_{i=1}^n (x_i-x)
   (y_i-y)}}
   {\displaystyle{\left[
   \sum_{i=1}^n(x_i-x)^2
   \sum_{i=1}^n(y_i-y)^2
   \right]^{1/2}}}
\end{equation*}
```

$$P = \frac{\displaystyle{\sum_{i=1}^n (x_i-x)(y_i-y)}}{\displaystyle{\left[\sum_{i=1}^n(x_i-x)^2\sum_{i=1}^n(y_i-y)^2\right]^{1/2}}}$$

Changer de style modifie également la façon dont les limites et les grands opérateurs sont affichés.

3.7.1 Symboles gras

Il est relativement ardu d'obtenir des symboles gras avec LaTeX ; cela est sans doute fait exprès car les typographes amateurs ont tendance à en abuser. La commande de changement de graisse `\mathbf` permet d'obtenir des caractères gras, mais romains (donc droits) alors que les symboles mathématiques sont normalement en italique. De plus elle ne fonctionne pas sur les minuscules grecques. Il y a bien une commande `\boldmath`, mais *elle ne peut être utilisée qu'en dehors du mode mathématique.* Cependant elle fonctionne aussi pour les symboles :

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

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

L'extension amsbsy (appelée par amsmath) ainsi que l'extenstion bm dans le paquet `tools` simplifient beaucoup ce problème puisqu'ils fournissent une commande `\boldsymbol` :

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

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

3.8 Théorèmes, lemmes, etc.

En rédigeant des documents mathématiques, on a besoin d'un moyen de présenter des lemmes, des définitions, des axiomes et d'autres structures similaires :

`\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 dans le document final.

Les arguments entre crochets sont optionnels. Ils servent à indiquer la numérotation à utiliser sur le théorème. Utilisez le *compteur* pour indiquer 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*`}`

L'extension amsthm (qui fait partie d'$\mathcal{A}\kern-.1667em\lower.5ex\hbox{$\mathcal{M}$}\kern-.125em\mathcal{S}$-LaTeX) met à disposition la commande `\theoremstyle{`*style*`}` offrant un choix de type de théorème parmi trois styles préféfinis : `definition` (titre gras, corps romain), `plain` (titre gras, corps italique) or `remark` (titre italique, corps romain).

Voilà pour la théorie. Les exemples qui suivent devraient lever tout doute et montrer clairement que la commande `\newtheorem` est trop complexe à comprendre.

D'abord définissez les théorèmes :

```
\theoremstyle{definition} \newtheorem{loi}{loi}
\theoremstyle{plain}      \newtheorem{decret}[loi]{Décret}
\theoremstyle{remark}     \newtheorem*{lechef}{Le chef}
```

```
\begin{loi} \label{chef}
Le chef a raison.
\end{loi}
\begin{decret}[Important]
Le chef a toujours raison,
voir la loi~\ref{chef}.
\end{decret}
\begin{lechef}
Et si le chef a tort, se référer
à la loi~\ref{chef}.
\end{lechef}
```

loi 1. Le chef a raison.

Décret 2 (Important)**.** *Le chef a toujours raison, voir la loi 1.*

Le chef. Et si le chef a tort, se référer à la loi 1.

L'environnement "decret" utilise le même compteur que l'environnement "loi", donc il obtient un numéro dans la même séquence que les autres "lois". L'argument entre crochets permet de spécifier un titre ou quelque chose de ce genre pour l'environnement.

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

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

Les environnements "Murphy" sont numérotés à l'intérieur de la section en cours. On aurait pu utiliser un autre niveau tel que `chapter` ou `subsection`.

Si vous voulez personnaliser vos théorèmes au point près, l'extension ntheorem vous offrira la pléthore d'options dont vous avez besoin.

3.8.1 Preuves et symbole de fin de preuve

L'extension amsthm fournit aussi l'environnement de preuve `proof`.

```
\begin{proof}
 Trivial, utilisez
 \begin{equation*}
   E=mc^2.
 \end{equation*}
\end{proof}
```

Proof. Trivial, utilisez

$$E = mc^2.$$

□

La commande `\qedhere` permet de déplacer le symbole de fin de preuve (CQFD) pour les cas où il finirait seul sur une ligne.

```
\begin{proof}
 Trivial, utilisez
 \begin{equation*}
   E=mc^2. \qedhere
 \end{equation*}
\end{proof}
```

Proof. Trivial, utilisez

$$E = mc^2. \qquad \square$$

Malheureusement, cette correction ne fonctionne pas avec l'environnement `IEEEeqnarray`.

```
\begin{proof}
  Voici une preuve qui termine
  par un alignement d'équations.
  \begin{IEEEeqnarray*}{rCl}
    a & = & b + c \\
    & = & d + e. \qedhere
  \end{IEEEeqnarray*}
\end{proof}
```

Proof. Voici une preuve qui termine par un alignement d'équations.

$$\begin{aligned} a &= b + c \\ &= d + e. \quad \square \end{aligned}$$

La raison provient de la structure interne d'`IEEEeqnarray`, qui place systématiquement une colonne invisible de chaque côté de l'équation, qui contiennent seulement des espaces étirables, de façon à ce que le tableau d'équations soit correctement centré. Il faudrait en fait placer le `\qedhere` à l'extérieur de cet espace étirable, mais ce n'est pas possible car ces colonnes sont invisibles pour l'utilisateur.

Il y a une solution très simple : définir ces colonnes étirables explicitement !

```
\begin{proof}
  Voici une preuve qui termine
  par un alignement d'équations.
  \begin{IEEEeqnarray*}{+rCl+x*}
    a & = & b + c \\
    & = & d + e. & \qedhere
  \end{IEEEeqnarray*}
\end{proof}
```

Proof. Voici une preuve qui termine par un alignement d'équations.

$$\begin{aligned} a &= b + c \\ &= d + e. \qquad\qquad \square \end{aligned}$$

Ici, le `+` dans `{+rCl+x*}` indique un espace étirable : un à gauche de l'équation qui, s'il n'était pas présent, serait ajouté d'office par `IEEEeqnarray`, et un à droite de l'équation. Mais cette fois à droite, *après* cette colonne étirable, nous ajoutons une colonne vide `x`, qui ne sera utilisée que sur la dernière ligne, si la commande `\qedhere` est utilisée. Enfin, nous ajoutons un `*` : c'est un espace sans épaisseur, qui empêche `IEEEeqnarray` d'ajouter un autre espace `+` indésirable.

Dans le cas d'équations numérotées, le problème est similaire. Comparons

```
\begin{proof}
  Voici une preuve qui termine
  par une équation numérotée.
  \begin{equation}
    a = b + c.
  \end{equation}
\end{proof}
```

Proof. Voici une preuve qui termine par une équation numérotée.

$$a = b + c. \tag{3.42}$$

$\square$

avec

```
\begin{proof}
  Voici une preuve qui termine
  par une équation numérotée.
  \begin{equation}
    a = b + c. \qedhere
  \end{equation}
\end{proof}
```

Proof. Voici une preuve qui termine par une équation numérotée.

$$a = b + c. \tag{3.43}$$

$\square$

Vous remarquerez que dans la deuxième version, qui est correcte, le symbole □ est bien plus proche de l'équation que dans la première.

De façon analogue, la bonne manière de placer le symbole de fin de preuve à la fin d'un alignements d'équations est la suivante.

```
\begin{proof}
  Voici une preuve qui termine
  par un alignement d'équations.
  \begin{IEEEeqnarray}{+rCl+x*}
    a & = & b + c \\
    & = & d + e. \\
    &&& \qedhere\nonumber
  \end{IEEEeqnarray}
\end{proof}
```

Proof. Voici une preuve qui termine par un alignement d'équations.

$$a = b + c \quad (3.44)$$

$$= d + e. \quad (3.45)$$

□

par opposition à

```
\begin{proof}
  Voici une preuve qui termine
  par un alignement d'équations.
  \begin{IEEEeqnarray}{rCl}
    a & = & b + c \\
    & = & d + e.
  \end{IEEEeqnarray}
\end{proof}
```

Proof. Voici une preuve qui termine par un alignement d'équations.

$$a = b + c \quad (3.46)$$

$$= d + e. \quad (3.47)$$

□

3.9 Liste des symboles mathématiques

Les tableaux suivants montrent tous les symboles accessibles en mode *mathématique*.

Remarquez que certains tableaux montrent des symboles qui ne sont accessibles qu'après avoir chargé l'extension `amssymb` dans le préambule [14]. Si les extensions et les polices de l'$\mathcal{AMS}$ ne sont pas installées sur votre système, vous pouvez les récupérer sur
`macros/latex/required/amslatex`. Il existe une liste beaucoup plus complète de symboles sur `info/symbols/comprehensive`.

Table 3.1: Accents en mode mathématique

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

Table 3.2: Alphabet grec

Certaines lettres n'ont pas leur équivalent en majuscule comme `\Alpha`, `\Beta`... parce qu'elles ressemblent aux lettres romaines normales : A, B...

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

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

Table 3.3: Relations binaires

Vous pouvez produire la négation de ces symboles en les préfixant par la commande `\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`

Table 3.4: 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

Table 3.5: Opérateurs n-aires

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

Table 3.6: Flèches

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

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

Table 3.7: Flèches en tant qu'accents

$\overrightarrow{AB}$	`\overrightarrow{AB}`	$\underrightarrow{AB}$	`\underrightarrow{AB}`
$\overleftarrow{AB}$	`\overleftarrow{AB}`	$\underleftarrow{AB}$	`\underleftarrow{AB}`
$\overleftrightarrow{AB}$	`\overleftrightarrow{AB}`	$\underleftrightarrow{AB}$	`\underleftrightarrow{AB}`

Table 3.8: Délimiteurs

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

Table 3.9: Grands délimiteurs

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

Table 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

Table 3.11: Symboles non-mathématiques

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

†	`\dag`	§	`\S`	©	`\copyright`	®	`\textregistered`
‡	`\ddag`	¶	`\P`	£	`\pounds`	%	`\%`

Table 3.12: Délimiteurs de l'$\mathcal{AMS}$

$\ulcorner$	`\ulcorner`	$\urcorner$	`\urcorner`	$\llcorner$	`\llcorner`	$\lrcorner$	`\lrcorner`

Table 3.13: Caractères grecs et hébreux de l'$\mathcal{AMS}$

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

Table 3.14: Alphabets mathématiques

Voir le tableau 6.4 page 127 pour d'autres polices mathématiques.

Exemple	Commande	Extension à utiliser
$\mathrm{ABCDEabcde1234}$	`\mathrm{ABCDE abcde 1234}`	
$\mathit{ABCDEabcde1234}$	`\mathit{ABCDE abcde 1234}`	
$\mathnormal{ABCDEabcde1234}$	`\mathnormal{ABCDE abcde 1234}`	
$\mathcal{ABCDE}$	`\mathcal{ABCDE abcde 1234}`	
$\mathscr{ABCDE}$	`\mathscr{ABCDE abcde 1234}`	mathrsfs
$\mathfrak{ABCDEabcde1234}$	`\mathfrak{ABCDE abcde 1234}`	amsfonts ou amssymb
$\mathbb{ABCDE}$⅁⊮⊭⊮⋭	`\mathbb{ABCDE abcde 1234}`	amsfonts ou amssymb

Table 3.15: Opérateurs binaires de l'$\mathcal{AMS}$

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

Table 3.16: Relations binaires de l'$\mathcal{AMS}$

$\lessdot$	`\lessdot`	$\gtrdot$	`\gtrdot`	$\doteqdot$	`\doteqdot`
$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`	$\risingdotseq$	`\risingdotseq`
$\eqslantless$	`\eqslantless`	$\eqslantgtr$	`\eqslantgtr`	$\fallingdotseq$	`\fallingdotseq`
$\leqq$	`\leqq`	$\geqq$	`\geqq`	$\eqcirc$	`\eqcirc`
$\lll$	`\lll` ou `\llless`	$\ggg$	`\ggg`	$\circeq$	`\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`
$\shortparallel$	`\shortparallel`	$\Supset$	`\Supset`	$\Vvdash$	`\Vvdash`
$\blacktriangleleft$	`\blacktriangleleft`	$\sqsupset$	`\sqsupset`	$\backepsilon$	`\backepsilon`
$\vartriangleright$	`\vartriangleright`	$\because$	`\because`	$\varpropto$	`\varpropto`
$\blacktriangleright$	`\blacktriangleright`	$\Subset$	`\Subset`	$\between$	`\between`
$\trianglerighteq$	`\trianglerighteq`	$\smallfrown$	`\smallfrown`	$\pitchfork$	`\pitchfork`
$\vartriangleleft$	`\vartriangleleft`	$\shortmid$	`\shortmid`	$\smallsmile$	`\smallsmile`
$\trianglelefteq$	`\trianglelefteq`	$\therefore$	`\therefore`	$\sqsubset$	`\sqsubset`

Table 3.17: Flèches de l'$\mathcal{AMS}$

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

Table 3.18: Négations des relations binaires et des flèches de l'$\mathcal{AMS}$

$\nless$	`\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`

Table 3.19: Symboles divers de l'$\mathcal{AMS}$

$\hbar$	`\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`		
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`	$\backprime$	`\backprime`
$\nexists$	`\nexists`	$\Finv$	`\Finv`	$\varnothing$	`\varnothing`
$\eth$	`\eth`	$\sphericalangle$	`\sphericalangle`	$\mho$	`\mho`

Chapter 4

Compléments

Pour rédiger un document important, LaTeX vous fournit des outils pour réaliser un index, une liste de références bibliographiques et d'autres choses. Des descriptions bien plus complètes de ces possibilités et des améliorations possibles avec LaTeX se trouvent dans le *LaTeX Manual* [1],dans *le LaTeX Companion* [3] et dans *LaTeX, apprentissage, guide et référence* [5].

4.1 Inclusion PostScript encapsulé

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

Il existe plusieurs façons de produire des graphiques avec des commandes LaTeX ou des extensions de LaTeX. Certaines d'entre elles sont décrites au chapitre 5. Pour plus d'informations à leur sujet, reportez-vous à *LaTeX, apprentissage, guide et référence* [5] ou à *le LaTeX Companion* [3].

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

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

1. Tel que XFig, Gnuplot, Gimp, Xara X...
2. Une autre possibilité pour imprimer du PostScript est d'utiliser le logiciel GhostScript, disponible sur `CTAN://support/ghostscript`. Les utilisateurs de Windows et OS/2 peuvent se tourner vers GSview.
3. `CTAN://macros/latex/required/graphics`.

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

1. exportez la figure de votre logiciel graphique au format EPS [4];
2. chargez l'extension `graphicx` dans le préambule de votre fichier source avec :

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

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

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

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

Table 4.1: Clefs pour l'extension `graphicx`

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

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

```
\begin{figure}
\centering
```

4. Si votre logiciel ne sait pas exporter au format EPS, vous pouvez essayer d'utiliser un pilote d'impression PostScript (tel que Apple LaserWriter, par exemple) puis imprimer vers un fichier à l'aide de ce pilote. Avec de la chance ce fichier sera au format EPS. Notez cependant qu'un fichier EPS ne doit pas faire plus d'une page. Certains pilotes d'impression peuvent même être configurés pour produire explicitement un format EPS.

```
\includegraphics[angle=90,
                 width=0.5\textwidth]{test}
\caption{Ceci est un test.}
\end{figure}
```

Cette commande inclut la figure stockée dans le fichier `test.eps`. La figure est *d'abord* tournée de 90 degrés puis ajustée pour que sa largeur finale soit de 10 cm. Les proportions largeur/hauteur sont conservées, puisqu'aucune hauteur n'est spécifiée.

Pour plus d'informations, reportez vous à [10] et [17].

4.2 Références bibliographiques

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

`\bibitem[`*label*`]{`*marque*`}`

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

`\cite{`*marque*`}`

Si vous n'utilisez pas l'option *label*, les références sont automatiquement numérotées. Le paramètre qui suit `\begin{thebibliography}` définit la largeur du décrochement utilisé pour placer ces numéros. Dans l'exemple ci-après, `{99}` indique à LaTeX que le décrochement ne devrait jamais être plus large que le nombre 99.

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

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

Partl [1] propose que...

Bibliography

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

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

4.3 Index

L'index est un élément fort utile pour de nombreux ouvrages. LATEX et le programme associé `makeindex`[5] permettent de créer des index assez facilement. Cette introduction présente seulement les commandes élémentaires de gestion d'un index. Pour une description plus détaillée, reportez-vous à *le LATEX Companion* [3]ou à *LATEX, apprentissage, guide et référence* [5].

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

```
\usepackage{makeidx}
```

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

```
\makeindex
```

placée dans le préambule.

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

`\index{`*clef@entrée formatée*`}`

où *entrée formatée* est ce qui doit apparaître dans l'index, et *clef* est utilisée pour le tri. La partie *entrée formattée* est optionnelle. Vous insérez des commandes `\index` aux endroits du texte que vous voulez voir référencés par l'index. Le tableau 4.2 explique cela 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 `.idx`. Ce fichier est ensuite traité par le programme `makeindex` :

`makeindex` *nom de fichier*

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

Table 4.2: Exemples de clefs d'index

Exemple	**Résultat**	**Commentaires**
`\index{hello}`	hello, 1	Entrée normale
`\index{hello!Peter}`	Peter, 3	Sous-entrée de 'hello'
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	Entrée formatée
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	Entrée formatée
`\index{Kaese@K\"ase}`	**Käse**, 33	Entrée formatée
`\index{ecole@\'ecole}`	école, 4	Entrée formatée
`\index{Jenny|textbf}`	Jenny, **3**	Numéro de page formaté
`\index{Joe|textit}`	Joe, *5*	Numéro de page formaté

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

```
\printindex
```

L'extension showidx fournie avec LaTeX 2_ε 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.

Remarquez également que la command `\index` peut affecter votre mise en page si vous n'y prenez pas garde.

```
Mon mot \index{mot}. Différent
de mot\index{mot}. Notez la
position du point final.
```

Mon mot . Différent de mot. Notez la position du point final.

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

4.4 En-têtes améliorés

L'extension fancyhdr[6], développée par Piet van Oostrum, offre quelques commandes simples permettant de personnaliser les entêtes et les pieds de page de votre document. Si vous regardez en haut de cette page, vous verrez un résultat possible de l'utilisation de cette extension.

6. disponible sur `CTAN://macros/latex/contrib/supported/fancyhdr`.

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

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

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

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

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

Figure 4.1: Exemple de configuration de l'extension fancyhdr

4.5 L'extension verbatim

Plus haut dans ce document, vous avez appris à utiliser l'*environnement* `verbatim`. Dans cette section vous allez découvrir l'*extension* verbatim. L'extension verbatim est une nouvelle implémentation de l'environnement du même nom qui corrige certaines de ses limitations. En soi cela n'est pas spectaculaire, mais ce package s'adjoint de nouvelles fonctionnalités qui justifient que cette extension soit citée ici. L'extension verbatim propose la commande :

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

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

Puisque l'extension verbatim fait partie de l'ensemble "`tools`", elle devrait être déjà disponible sur la plupart des systèmes. Pour en savoir plus au sujet de cette extension, reportez-vous à [11].

4.6 Installation d'extensions

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

Les extensions, telles geometry, hyphenat et beaucoup d'autres, sont en général fournies sous la forme de deux fichiers, l'un de suffixe `.dtx`, l'autre de suffixe `.ins`. Souvent un fichier `readme.txt` leur est joint et donne une brève description de l'extension. Le mieux est alors de commencer par la lecture de ce fichier.

Quoi qu'il en soit, une fois que vous avez copié les fichiers de l'extension sur votre machine, vous dever les manipuler de manière à (a) informer votre distribution TeX de cette nouvelle extension et (b) obtenir sa documentation. Voici la manière de procéder :

1. exécuter LaTeX sur le fichier `.ins`. Ceci produira les fichiers `.sty`, `.def`, etc., dont LaTeX a besoin.

2. déplacer ces fichiers dans un répertoire adéquat, en général c'est dans `.../texmf/tex/latex` ou dans `.../`*`localtexmf`*`/tex/latex`.

3. mettre à jour la base de données des noms de fichiers, la commande dépend de votre distribution LaTeX : TeXlive – `texhash`; web2c – `mktexlsr`; MiKTeX – `initexmf --update-fndb` ou via l'interface graphique; la commande peut également être `texconfig rehash`.

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

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

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

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

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

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

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

4.7 Travailler avec pdfLaTeX

Par Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

PDF est un format de document hypertexte et portable. De la même manière que dans une page Web, certains mots sont marqués comme des hyperliens. Ils renvoient vers d'autres endroits du document voir vers d'autres documents. En cliquant sur un hyperlien vous serez transportés sur la destination de ce lien. Dans le contexte de LaTeX, cela signifie que toute occurrence de `\ref` et de `\pageref` peut devenir un hyperlien. De plus, la table des matières, l'index et d'autres structures similaires deviendront aussi des collections d'hyperliens.

La plupart des pages Web de nos jours sont écrites en HTML *(HyperText Markup Language)*. Ce format a deux défauts majeurs pour écrire des documents scientifiques :

1. L'inclusion de formules mathématiques n'est généralement pas possible. Bien qu'il y ait un standard pour cela, la plupart des navigateurs ne le prennent pas en compte ou n'ont pas les polices requises;
2. L'impression de documents HTML est possible mais les résultats varient énormément selon la plateforme et le navigateur, et loin des standards de qualité du monde LaTeX.

Il y a de nombreuses tentatives de création de traducteurs de LaTeX vers HTML. Certaines rencontrèrent un certain succès dans le sens où elles peuvent produire des pages Web lisibles à partir d'un fichier d'entrée LaTeX standard. Mais toutes évitent certaines parties délicates pour obtenir ce résultat. Dès que l'on utilise des fonctionnalités plus complexes de LaTeX ou des extensions externes, les choses tendent à partir à vau-l'eau. Les auteurs qui souhaitent préserver la qualité typographique unique de leurs documents, même sur le Web, se tournent vers le format PDF *(Portable Document Format)* qui préserve la mise en page du document et autorise la navigation hypertextuelle. La plupart des navigateurs disposent de plus d'extensions pour l'affichage direct de documents PDF.

Bien qu'il y ait des visualiseurs DVI et PS pour presque toutes les plateformes, vous constaterez qu'Acrobat Reader et Xpdf, qui servent à lire des documents PDF, sont encore plus fréquemment déployés[7]. Fournir des versions PDF de vos documents les rendra donc d'autant plus accessibles à vos lecteurs potentiels.

4.7.1 Des documents PDF pour le Web

La création d'un fichier PDF à partir d'une source LaTeX est très simple, grâce au programme pdfTeX développé par Hàn Thế Thành. pdfTeX produit une sortie PDF là où TeX produisait une sortie DVI. Il existe aussi pdfLaTeX qui produit une sortie PDF à partir de sources LaTeX.

pdfTeX et pdfLaTeX sont installés automatiquement par les distributions TeX modernes majeures : teTeX, fpTeX, MikTeX, TeXLive et CMacTeX.

Pour produire un fichier PDF au lieu d'un fichier DVI, il suffit de remplacer la commande `latex document.tex` par `pdflatex document.tex`. Sur les systèmes où LaTeX n'est pas appelé depuis la ligne de commande, vous pourrez trouver un bouton à cette fin dans l'interface graphique TeX.

Vous pouvez définir le format de papier via une option passée à `documentclass` comme `a4paper` ou `letterpaper`. Ceci fonctionne également avec pdfLaTeX, mais en plus de cela pdfTeX a besoin de connaître le format physique du papier pour déterminer la taille réelle des pages dans le fichier pdf. Si vous utilisez l'extension `hyperref` (voir page 97), le format de papier sera ajusté automatiquement. Sinon vous devrez le spécifier manuellement en ajoutant les lignes suivantes en préambule :

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

La section suivante précisera plus encore les différences entre LaTeX et pdfLaTeX. Les différences principales concernent trois domaines : les polices à utiliser, les formats d'images à inclure et la configuration manuelle des hyperliens.

4.7.2 Les polices

pdfLaTeX est capable de gérer toutes sortes de polices (PK bitmaps, TrueType, PostScript type 1...) mais le format de police normal de LaTeX, les polices bitmap PK, produisent des résultats exécrables lorsqu'ils sont affichés

7. `http://pdfreaders.org`

avec Acrobat Reader. Il est préférable d'utiliser des polices PostScript de Type 1 exclusivement pour produire des documents au rendu correct. *Les installations modernes de TEX sont réglées pour que ce comportement soit activé automatiquement. Le mieux est d'essayer. Si cela fonctionne pour vous, passez directement à la section suivante.*

L'ensemble de polices de Type 1 la plus utilisée aujourd'hui s'appelle Latin Modern (LM). Si vous avez une installation TEX récente, il est probable que vous en ayez déjà une copie installée. Tout ce qu'il vous reste alors à faire dans ce cas est d'ajouter

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

au préambule de votre document et vous êtes parés pour créer d'excellents documents PDF avec support complet de l'alphabet latin. Si vous travaillez avec une installation TEX réduite, vous aurez peut-être à ajouter les polices lm explicitement.

Pour la langue russe, une solution est d'utiliser des polices C1 virtuelles disponibles sur `ftp://ftp.vsu.ru/pub/tex/font-packs/c1fonts`. Ces polices combinent les polices standard CM type 1 de Bluesky aux polices type 1 CMCYR des collections Paradissa et BaKoMa, toutes disponibles sur le CTAN. Comme les polices Paradissa ne contiennent que des lettres russes, il manque aux polices C1 les autres glyphes cyrilliques.

Une autre solution est de passer à d'autres polices PostScript type 1. Certaines d'entre elles sont même fournies avec chaque copie d'Acrobat Reader. Celles-ci ayant des tailles de caractères différentes, l'aspect de votre texte sur vos pages sera différent. Généralement ces polices utiliseront plus d'espace que les polices CM qui sont très efficaces en terme d'occupation d'espace. De plus la cohérence visuelle de votre document en souffrira car les polices Times, Helvetica et Courier (les premières candidates en lice pour un tel changement) n'ont pas été conçues pour fonctionner en harmonie dans un seul document.

Deux polices prêtes à l'emploi sont disponibles dans cette optique : pxfonts, basée sur *Palatino* pour la police du texte principal et l'extension txfonts basée sur *Times*. Pour les utiliser il suffit d'ajouter les lignes suivantes en préambule :

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

Il se peut que vous voyiez des lignes comme

```
Warning: pdftex (file eurmo10): Font eur... not found
```

dans le fichier `.log` après compilation de votre fichier. Elles signifient que l'une des polices utilisées par votre document est introuvable. Assurez-vous d'identifier et réparer les sections incriminées de votre document, car le document PDF résultant pourrait *ne pas afficher du tout les pages contenant les caractères manquants.*

4.7.3 Images et graphiques

Inclure des images dans un document fonctionne mieux avec l'extension graphicx (voir page 87).

```
\usepackage{xcolor,graphicx}
```

L'exemple ci-dessus charge aussi l'extension xcolor, puisqu'il est naturel d'utiliser de la couleur dans les documents affichés à l'écran.

Voilà pour les bonnes nouvelles. La mauvaise nouvelle est que les figures au format PostScript encapsulé ne fonctionnent *pas* avec pdfLaTeX. Si vous ne donnez pas d'extension de fichier dans l'argument de la commande `\includegraphics`, graphicx cherchera lui-même la bonne en fonction de l'option de pilote qui lui est passée. Pour `pdftex` ce seront les formats `.png`, `.pdf`, `.jpg` et `.mps` (METAPOST), mais *sûrement pas* `.eps`.

La manière la plus simple de résoudre le problème est de convertir vos fichiers EPS au format PDF en utilisant l'utilitaire `epstopdf` disponible sur de nombreux systèmes. Pour les images vectorielles (dessins) c'est une bonne solution. Pour les formats bitmap (photos, numérisations) ce n'est pas l'idéal car le format PDF prend nativement en compte l'inclusion d'images PNG et JPEG. PNG est un bon format pour les captures d'écran et autres images contenant peu de couleurs. JPEG est très bien pour les photos de par l'efficacité de sa compression pour ce type d'images.

Pour certaines figures géométriques, il peut même être préférable d'utiliser un langage de commandes spécialisé comme METAPOST, qui se trouve dans la plupart des distributions TeX et est livré avec son manuel détaillé.

4.7.4 Liens hypertextuels

L'extension hyperref [8] prendra soin de transformer toutes les références internes de votre document en hyperliens. Pour que cela fonctionne, il faut faire appel à un peu de magie : vous devrez ajouter `\usepackage{hyperref}` en tant que *dernière* commande de votre préambule. De nombreuses options modifient le comportement de l'extension hyperref :

- soit en tant qu'une liste séparée par des virgules après l'option pdftex `\usepackage{hyperref}` ;
- soit sur des lignes à part avec la commande `\hypersetup{`*options*`}`.

Dans la liste suivante, les valeurs par défaut des options sont affichées avec une police droite.

`bookmarks (=true,`*`false`*`)` montrer ou cacher les marque-pages lors de l'affichage du document ;

`unicode (=false,`*`true`*`)` permettre d'utiliser des caractères non-latins dans les marque-pages du lecteur de PDF ;

8. Il est ici intéressant de noter que l'extension hyperref n'est pas limitée à pdfTeX. Elle peut aussi être configurée pour embarquer des informations spécifiques au format PDF dans la sortie DVI de LaTeX, qui seront insérées dans le fichier PS via `dvips` et enfin reconnues par l'interpréteur PostScript lors de la transformation du fichier PS en PDF.

pdftoolbar (=true,*false*) montrer ou cacher la barre d'outils du lecteur de PDF ;

pdfmenubar (=true,*false*) montrer ou cacher le menu du lecteur de PDF ;

pdffitwindow (=false,*true*) ajuster le grossissement initial lors de l'affichage du fichier PDF ;

pdftitle (={texte}) définir le titre affiché dans la fenêtre `Document Info` du lecteur de PDF ;

pdfauthor (={texte}) le nom de l'auteur ;

pdfnewwindow (=false,*true*) définir si une nouvelle fenêtre doit être ouverte lorsqu'un lien conduit hors du document courant ;

colorlinks (=false,*true*) entourer les liens par des liserés colorés (`false`) ou colorer le texte des liens (`true`). La couleur de ces liens peut être configurée via les options suivantes (les couleurs par défaut sont indiquées) :

  linkcolor (=red) couleur des liens internes (sections, pages, etc) ;

  citecolor (=green) couleur des liens de citations bibliographiques ;

  filecolor (=magenta) couleur des liens vers des fichiers ;

  urlcolor (=cyan) couleur des liens URL (adresse électronique, adresse web).

Si les valeurs par défaut vous plaisent, utilisez

```
\usepackage{hyperref}
```

Pour avoir la liste des marque-pages ouverte et des liens en couleur (les valeurs `=true` sont optionnelles) :

```
\usepackage[bookmarks,colorlinks]{hyperref}
```

Lors de la création de fichiers PDFs en vue d'impression, les liens colorés peuvent finir grisés dans le résultat final, ce qui les rend difficiles à lire. Vous pouvez utiliser des liserés colorés qui ne seront pas imprimés :

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

ou noircir les liens :

```
\usepackage{hyperref}
\hypersetup{colorlinks,
            citecolor=black,
            filecolor=black,
            linkcolor=black,
            urlcolor=black,
            }
```

Lorsque vous souhaitez simplement fournir des informations en section `Document Info` pour le fichier PDF :

```
\usepackage[pdfauthor={Pierre Desproges},
            pdftitle={Des femmes qui tombent},
            ]{hyperref}
```

En plus des hyperliens automatiques pour les références croisées, il est possible d'insérer des liens explicites via

`\href{`*url*`}{`*text*`}`

Le code

```
Le site web du \href{http://www.ctan.org}{CTAN}.
```

produit la sortie "CTAN"; cliquer sur le mot "CTAN" vous amènera au site web du CTAN.

Si la destination du lien n'est pas une URL mais un fichier local, vous pouvez utiliser la commande `\href` sans la partie 'http://' :

```
Le document complet est \href{manuel.pdf}{ici}
```

Elle produit le texte "Le document complet est ici". Cliquer sur le mot "ici" ouvrira le fichier `manuel.pdf` (le nom de fichier est relatif au document courant).

L'auteur d'un article peut souhaiter que ses lecteurs puissent lui envoyer facilement des emails en utilisant la commande `\href` à l'intérieur de la commande `\author` sur la page de titre du document :

```
\author{Mary Oetiker <\href{mailto:mary@oetiker.ch}%
       {mary@oetiker.ch}>
```

Remarquez que j'ai écrit le lien de manière à ce que mon adresse email apparaisse non seulement dans le lien mais aussi sur la page elle-même. J'ai fait cela parce que le lien
`\href{mailto:mary@oetiker.ch}{Mary Oetiker}`,
s'il fonctionnerait bien en listant le PDF, ne serait plus visible une fois la page imprimée.

4.7.5 Problème de liens

Des messages tels que celui-ci :

```
! pdfTeX warning (ext4): destination with the same
  identifier (name{page.1}) has been already used,
  duplicate ignored
```

apparaissent lors de la réinitialisation d'un compteur, par exemple lors de l'utilisation de la commande `\mainmatter` fournie par la classe de document `book`. Elle remet à 1 le numéro de page page avant le premier chapitre du livre. Cependant, la préface du livre a aussi une page numérotée 1 : les liens vers la "page 1" ne seraient alors plus uniques, d'où l'avertissement "`duplicate` has been `ignored`".

La contre-mesure consiste en l'ajout de `plainpages=false` aux options d'hyperref. Ça ne résout le problème que pour le compteur de pages, malheureusement. Une solution encore plus radicale est d'utiliser l'option `hypertexnames=false` mais cela empêchera les liens de pages en index de fonctionner.

4.7.6 Problèmes de marque-pages

Le texte affiché par les marque-pages ne correspond pas toujours à vos attentes : comme ceux-ci sont "juste du texte", encore moins de caractères que pour LaTeX sont disponibles. Hyperref remarquera ce type de problème et notifiera un avertissement :

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

Vous pouvez contourner ce problème en fournissant une chaîne de caractères pour les marque-pages pour remplacer le texte à problèmes :

`\texorpdfstring{`*Texte TeX*`}{`*Texte de marque-pages*`}`

Les expressions mathématiques sont les coupables idéales pour ce genre de problèmes :

```
\section{\texorpdfstring{$E=mc^2$}%
        {E=mc**2}}
```

transforme `$E=mc^2$` en "E=mc**2" dans la zone de marque-pages.

Si votre document est écrit en Unicode et que vous utilisez l'option `unicode` de l'extension `hyperref`, vous pouvez utiliser des caractères Unicode dans les marque-pages. Vous aurez ainsi une bien plus grande sélection parmi les caractères utilisables avec `\texorpdfstring`.

4.7.7 Compatibilité des sources entre LaTeX et pdfLaTeX

Idéalement vos documents devraient se compiler aussi bien avec LaTeX qu'avec pdfLaTeX. Le problème principal ici est l'inclusion d'images. La solution simple est de *systématiquement omettre* le suffixe de fichier des commandes `\includegraphics`. Ils chercheront alors un fichier dans un format pris en compte dans le répertoire courant. Tout ce que vous avez à faire est créer des versions appropriées des images. LaTeX cherchera les fichiers `.eps` et pdfLaTeX s'efforcera d'inclure un fichier au suffixe `.png`, `.pdf`, `.jpg` ou `.mps` (dans cet ordre).

Pour les occasions où vous voulez utiliser un code différent pour la version PDF de votre document, vous pouvez ajouter l'extension `ifpdf` [9] en préambule. Il y a des chances qu'elle soit déjà installée ; si ce n'est pas le cas alors

9. Pour le pourquoi de l'utilisation de cette extension, consultez la FAQ TeX (en anglais) à la page
`http://www.tex.ac.uk/cgi-bin/texfaq2html?label=ifpdf`.

vous utilisez probablement MiKTEX qui l'installera de lui-même lorsque vous l'utiliserez pour la première fois. Cette extension définit la commande spéciale `\ifpdf` qui permet d'écrire du code conditionnel aisément. Dans l'exemple suivant, nous voulons que la version PostScript soit en noir et blanc à cause des coûts d'impression, mais nous voulons quand même que la version PDF en ligne soit colorée.

```
\RequirePackage{ifpdf} % Production d'un PDF ?
\documentclass[a4paper,12pt]{book}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[bookmarks, % ajout d'hyperliens
            colorlinks,
            plainpages=false]{hyperref}
\usepackage{graphicx}
\ifpdf
  \hypersetup{linkscolor=blue}
\else
  \hypersetup{linkscolors=black}
\fi
\usepackage[english]{babel}
...
```

L'exemple ci-dessus fait inclure l'extension hyperref même dans la version non-PDF, ceci afin de faire fonctionner la commande `\href` dans tous les cas. Cela évite de devoir protéger chacune de ses occurrences par une structure conditionnelle.

Remarquez que dans les distributions TEX récentes (comme TEXLive, MacTEX et MiKTEX), le programme TEX par défaut est en fait pdfTEX. Il s'adaptera automatiquement à la production de pdf ou de dvi en fonction de la façon dont vous l'appelez : utilisez la commande `pdflatex` pour obtenir une sortie pdf et `latex` pour une sortie dvi.

4.8 Utiliser XƎLATEX

Par Axel Kielhorn <A.Kielhorn@web.de>

La plupart des propos applicables à pdfLATEX sont également valides pour XƎLATEX.

Le Wiki situé sur le site `http://wiki.xelatex.org/doku.php` rassemble les informations concernant XƎTEX et XƎLATEX.

4.8.1 Les polices

En plus des polices usuelles basées sur `tfm`, XƎLATEX sait utiliser toute police connue du système d'exploitation. Supposons que les polices `Linux Libertine` soient installées, vous pouvez les utiliser via la commande

```
\usepackage{fontspec}
\setmainfont[Ligatures=TeX]{Linux Libertine}
```

ajoutée dans le préambule. La détection des versions italique et grasse de la police devrait également être effective, ainsi `\textit` et `\textbf` fonctionneront comme de coutume. Lorsque la police utilise la technologie OpenType, vous avez accès à de nombreuses fonctionnalités qui nécessitaient naguère le passage à une autre police ou l'utilisation de polices virtuelles. La fonctionnalité principale est l'obtention d'un ensemble de caractères étendu : une police peut contenir des caractères latins, grecs et cyrilliques ainsi que les ligatures correspondantes.

De nombreuses polices contiennent au moins deux types de chiffres, ceux à l'alignement normal et ceux en style ancien (ou en minuscules) qui s'étendent partiellement en dessous de la ligne. Elles peuvent contenir des chiffres proportionnés (le "1"prend moins de place que le "0") ou à espacement fixe mieux adaptés pour des tableaux.

```
\newfontfamily\LLln[Numbers=Lining]{(font)}
\newfontfamily\LLos[Numbers=OldStyle]{(font)}
\newfontfamily\LLlnm[Numbers=Lining,Numbers=Monospaced]{(font)}
\newfontfamily\LLosm[Numbers=OldStyle,Numbers=Monospaced]{(font)}
```

Presque toutes les polices OpenType contiennent les ligatures standard (fl fi ffi) mais il existe aussi des ligatures rares ou historiques comme st, ct et tz. Si leur utilité dans un rapport technique est discutable, elles ont leur place dans un roman. Pour activer ces ligatures utilisez l'une des commandes suivantes :

```
\setmainfont[Ligatures=Rare]{(font)}
\setmainfont[Ligatures=Historic]{(font)}
\setmainfont[Ligatures=Historic,Ligature=Rare]{(font)}
```

Toutes les polices ne contiennent pas les deux ensembles de ligatures, aussi consultez la documentation de la police ou essayez-la directement. Parfois ces ligatures dépendent du langage utilisé, par exemple la ligature (fk), inconnue en français, est utilisée en polonais. Il vous faudra ajouter

```
\setmainfont[Language=Polish]{(font)}
```

pour activer les ligatures polonaises.

Certaines polices (comme la police commerciale Adobe Garamond Premier Pro) contiennent aussi des glyphes alternatifs activés par défaut dans le XƎLaTeX fourni avec TeXLive 2010 [10]. Le résultat de son utilisation est par exemple un "Q"majuscule stylisé dont la queue rejoint par en-dessous le "u"qui suit. Pour désactiver cette fonctionnalité, vous devrez définir la police sans les traitements contextuels :

```
\setmainfont[Contextuals=NoAlternate]{(font)}
```

10. Ce comportement n'était pas présent car désactivé par défaut dans des versions précédentes.

Pour en savoir plus sur les polices et XƎLaTeX, référez-vous au manuel de fontspec.

Où se procurer des polices OpenType ?

Si `TeXLive` est installé, vous en avez quelques-unes à disposition dans `.../texmf-dist/fonts/opentype` : il ne vous reste plus qu'à les installer dans votre système d'exploitation. Cette collection ne comprend pas `DejaVu`, qui elle est disponible sur le site `http://dejavu-fonts.org/`.

Assurez-vous que chacune des polices n'est bien installée *qu'une seule fois*, sinon vous obtiendrez des résultats au mieux "intéressants".

Vous pouvez utiliser toute police installée sur votre ordinateur, rappelez-vous cependant que d'autres utilisateurs ne les ont peut-être pas. Par exemple, la police Zapfino utilisée dans le manuel de fontspec existe sous Mac OSX mais pas sous Windows. [11]

Saisie de caractères Unicode

Le nombre de caractères dans une police a assurément augmenté, mais ce n'est pas le cas du nombre de touches des claviers. Comment, alors, saisir des caractères non-ASCII ?

Si vous écrivez beaucoup de texte dans une autre langue, vous pouvez installer un clavier pour ce langage et imprimer les positions des caractères (la plupart des systèmes d'exploitation ont une forme ou une autre de clavier virtuel, il suffit de faire une copie d'écran).

Si vous n'avez pas souvent besoin de caractères exotiques, vous pouvez simplement le choisir dans la palette des caractères.

Certains environnements (comme p.e. le système X Window) offrent des méthodes spécifiques pour entrer des caractères non-ASCII. Il en va de même de certains éditeurs (comme Vim et Emacs). Consultez les documentations des outils que vous utilisez.

4.8.2 Compatibilité entre XƎLaTeX et pdfLaTeX

Certaines différences existent entre XƎLaTeX et pdfLaTeX.

- Un document XƎLaTeX doit être écrit en Unicode (UTF-8) alors qu'un document pdfLaTeX peut utiliser d'autres types de codages.
- L'extension microtype ne fonctionne pas encore avec XƎLaTeX et le support pour la saillie de caractères (technique connue sous le nom de "character protrusion"en anglais) est en cours de développement.
- Tout ce qui a trait aux polices doit être vérifié (sauf si vous vous cantonnez à la police Latin Modern).

11. À noter qu'une version commerciale appelée Zapfino Extra existe.

4.9 Créer des présentations

Par Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

Vous pouvez présenter les résultats de vos travaux sur un tableau noir, avec des transparents ou directement depuis votre ordinateur portable grâce à un logiciel de présentation.

pdfLaTeX combiné à la classe beamer permettent la création de présentations en PDF très semblables à ce que vous feriez avec LibreOffice ou PowerPoint dans un bon jour, mais bien plus portable puisque des lecteurs PDF sont disponibles sur bien plus de systèmes.

La classe beamer utilise graphicx, color et hyperref avec des options adaptées aux présentations sur écrans.

À la compilation du code de la figure 4.2 [12] avec pdfLaTeX, vous obtiendrez un fichier avec une page de titre et une deuxième page dont les points seront révélés un à un lorsque vous avancez dans la présentation.

Un avantage notable de la classe beamer est sa capacité à produire un fichier PDF utilisable directement sans avoir à passer par une étape PostScript comme avec prosper ou sans demander un post-traitement comme les présentations créées avec l'extension ppower4.

Avec la classe beamer vous pouvez produire plus versions (modes) de votre document à partir de votre fichier d'entrée. Celui-ci peut contenir des instructions spéciales pour les différents modes entre chevrons (crochets obliques `<` et `>`). Les modes suivants sont disponibles :

beamer pour une présentation PDF comme au-dessus ;

trans pour les transparents ;

handout pour la version imprimée.

Le mode par défaut est `beamer`. Vous pouvez le changer via les options globales de la classe, comme `\documentclass[10pt,handout]{beamer}` pour obtenir les notes associées.

L'aspect de la présentation dépend du thème choisi. Vous pouvez soit utiliser les thèmes livrés avec la classe beamer, soit en créer un. Pour plus d'information, voyez la documentation dans `beameruserguide.pdf`.

Observons maintenant le code de la figure 4.2 de plus près.

Pour la version à l'écran de la présentation (`\mode<beamer>`) nous avons choisi le thème *Goettingen*. Celui-ci propose un panneau de navigation intégré à la table des matières. Les options nous permettent de choisir la taille de ce panneau (22 mm dans notre cas) et sa position (à droite du corps du texte). L'option *hideothersubsections* montre les titres de sections, mais uniquement les sous-sections de la section en cours. Il n'y a pas d'autre réglage spécifique pour `\mode<trans>` et `\mode<handout>`, ils apparaîtront selon leur mise en page usuelle.

12. Attention, l'exemple suppose un codage d'entrée en utf8, voyez la ligne d'appel de l'extension `inputenc`. (NdT)

```
\documentclass[10pt]{beamer}
\usepackage[utf8x]{inputenc}
\mode<beamer>{%
  \usetheme[hideothersubsections,
            right,width=22mm]{Goettingen}
}

\title{Une simple présentation}
\author[D. Flipo]{Daniel Flipo}
\institute{U.S.T.L. \& GUTenberg}
\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

\begin{document}

\begin{frame}<handout:0>
  \titlepage
\end{frame}

\section{Un exemple}

\begin{frame}
  \frametitle{Choses à faire un dimanche après-midi}
  \begin{block}{On peut \ldots}
    \begin{itemize}
      \item sortir le chien\dots \pause
      \item lire un livre\pause
      \item rendre fou un chat\pause
    \end{itemize}
  \end{block}
  et bien d'autres choses
\end{frame}
\end{document}
```

Figure 4.2: Exemple de code pour la classe beamer

Les commandes `\title{}`, `\author{}`, `\institute{}`, et `\titlegraphic{}` permettent de remplir la page de titre. Les arguments optionnels de `\title[]{}` et `\author[]{}` servent à spécifier une version spéciale du titre et de l'auteur (respectivement) qui apparaîtront sur le panneau fourni par le thème *Goettingen*.

Les titres et sous-titres du panneau proviennent de l'utilisation des commandes usuelles `\section{}` et `\subsection{}`. Celles-ci doivent être placées *en dehors* de l'environnement `frame`.

Les petites icônes de navigation au bas de l'écran permettent elles aussi de naviguer dans le document. Leur présence n'est pas dépendante du thème choisi.

Le contenu de chaque transparent ou chaque écran doit être placé dans un environnement `frame`. L'option entre chevrons ((< et >) nous permet de supprimer un cadre en particulier de l'une des versions de la présentation. Dans notre exemple, la première page n'apparaîtra pas dans la version imprimée à cause de l'option <handout:0>.

Il est particulièrement recommandé de donner un titre à chaque transparent, en plus du transparent de titre. Ce titre est fourni via la commande `\frametitle{}`. Pour obtenir un sous-titre, utilisez `\framesubtitle{}` ou une commande `block` comme dans l'exemple. Remarquez également que le contenu des commandes `\section{}` et `\subsection{}` n'apparaît pas sur le transparent.

La commande `\pause` dans l'environnement itemize fait que les points vont se révéler un à un. D'autres effets de présentations sont disponibles via les commandes `\only`, `\uncover`, `\alt` et `\temporal`. Vous pouvez aussi utiliser les chevrons à de nombreux endroits pour personnaliser encore plus votre présentation.

Dans tous les cas, nous vous recommandons fortement de lire la documentation de la classe beamer `beameruserguide.pdf` pour tout savoir sur les possibilités non évoquées ici. Cette classe est développée activement, consultez son site (`http://latex-beamer.sourceforge.net/`) pour plus d'informations récentes.

Chapter 5

Produire des graphiques mathématiques

De nombreuses personnes utilisent LaTeX pour mettre en page leur texte. Et puisque l'approche "orientée structure" est si commode, LaTeX fournit également, quoique limitées, des possibilités de production graphique sur la base de descriptions textuelles. De nombreuses extensions ont été créées pour surmonter ces limitations. Vous découvrirez quelques-unes d'entre elles dans cette section.

5.1 Vue d'ensemble

La création de graphiques avec LaTeX a une longue histoire. Celle-ci commence par l'environnement `picture` qui permet la création de graphiques par le placement d'éléments prédéfinis sur un canevas. Vous pourrez en trouver une description complète dans le *LaTeX Manual* [1]. Cet environnement fournit la commande `\qbezier`, le "`q`"signifiant "quadratique". De nombreuses courbes usuelles comme les cercles, ellipses ou caténaires peuvent être approchés de manière satisfaisante par des courbes quadratiques de Bézier, bien que cela requiert quelque labeur mathématique. Si de plus un langage de programmation externe est utilisé pour générer les blocs `\qbezier` des fichiers d'entrée, l'environnement `picture` devient soudain plus puissant.

Bien que programmer des images directement en LaTeX soit limité et souvent fatiguant, il reste des raisons pour continuer à le faire : en effet les documents produits ainsi sont de petite taille (en octets) et il n'y a pas de fichier graphique à adjoindre.

Cela était la situation jusqu'à il y a quelques années lorsque Till Tantau, célèbre pour l'extension beamer, proposa le Format de Graphiques Portable pgf et son extension associée TikZ tikz. Ce système permet de créer des graphiques vectoriels de haute qualité avec un support pour tous les systèmes TeX actuels, y compris ceux utilisant pdf comme format de prédilection.

À partir de ces nombreuses extensions de base, d'autres extensions furent développées pour des besoins spécifiques. Nombre d'entre elles sont décrites en détail dans *The LaTeX Graphics Companion* [4].

Le plus puissant outil graphique lié à LaTeX est sûrement METAPOST. Il s'agit d'une application tierce basée sur METAFONT par Donald E. Knuth. METAPOST possède le langage mathématique puissant et sophistiqué de METAFONT. À la différence de METAFONT qui génère des bitmaps, METAPOST génère du PostScript encapsulé, qui peut être importé dans LaTeX et même pdfLaTeX. Pour une présentation, voyez *A User's Manual for METAPOST* [19] ou le tutoriel de [21].

Vous pourrez trouver une discussion très détaillé des stratégies LaTeX et TeX pour les images (et les polices) dans *TeX Unbound* [20].

5.2 L'extension picture

Par Urs Oswald <osurs@bluewin.ch>

Comme indiqué plus haut, l'environnement `picture` fait partie du LaTeX standard et fonctionne très bien pour des tâches simples ou si vous souhaitez plus de contrôle sur la position exacte des éléments sur une page. Mais pour tout travail graphique complexe, vous devriez plutôt regarder du côté de TikZ, présenté en section 5.3 à la page 117.

5.2.1 Commandes de base

L'une des deux commandes [1] suivantes crée un environnement `picture` :

```
\begin{picture}(x, y)...\end{picture}
```

ou

```
\begin{picture}(x, y)(x0, y0)...\end{picture}
```

Les nombres x, y, x_0, y_0 se rapportent à une unité `\unitlength` qui peut être réinitialisée à tout moment (sauf à l'intérieur d'un environnement `picture`) avec une commande telle que

```
\setlength{\unitlength}{1.2cm}
```

La valeur par défaut d'`\unitlength` est `1pt`. La première paire (x, y) réserve un espace rectangulaire pour l'image dans le document. La deuxième paire optionnelle associe des coordonnées arbitraires au coin inférieur gauche du rectangle réservé.

1. Croyez-le ou non, l'environnement picture fonctionne directement, sans avoir à charger quelque extensions LaTeX 2_ε supplémentaire que ce soit.

La plupart des commandes de dessin suivent l'une des deux formes suivantes :

\put$(x, y)${*objet*}

ou

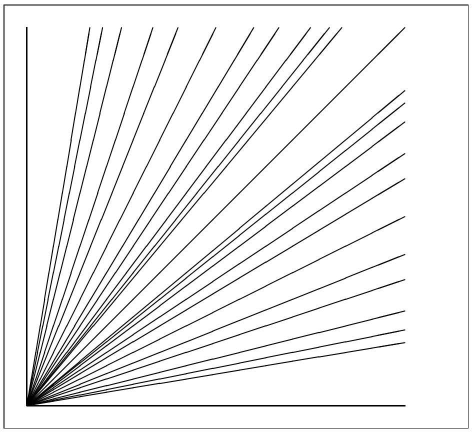
\multiput$(x, y)(\Delta x, \Delta y)${$n$}{*objet*}

Les courbes de Bézier font exception : elles sont dessinées avec la commande

\qbezier$(x_1, y_1)(x_2, y_2)(x_3, y_3)$

5.2.2 Segments

```
\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}
```



Les segments sont dessinés via

\put$(x, y)${\line$(x_1, y_1)${*longueur*}}

La commande `\line` a deux arguments :

1. Un vecteur de direction ;
2. Une longueur.

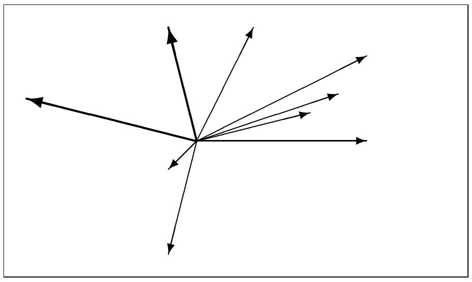
Les composants du vecteur de direction sont restreints aux entiers

$$-6, -5, \ldots, 5, 6,$$

et doivent être premiers entre eux (pas de diviseur en commun sauf 1). La figure ci-dessus illustre les 25 valeurs possibles de pentes pour le premier quadrant. La longueur est relative à `\unitlength`. L'argument de longueur est la coordonnée verticale dans le cas d'un segment vertical, horizontale dans tous les autres cas.

5.2.3 Flèches

```
\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
  \put(30,20){\vector(1,0){30}}
  \put(30,20){\vector(4,1){20}}
  \put(30,20){\vector(3,1){25}}
  \put(30,20){\vector(2,1){30}}
  \put(30,20){\vector(1,2){10}}
  \thicklines
  \put(30,20){\vector(-4,1){30}}
  \put(30,20){\vector(-1,4){5}}
  \thinlines
  \put(30,20){\vector(-1,-1){5}}
  \put(30,20){\vector(-1,-4){5}}
\end{picture}
```



Les flèches sont dessinées via

$\boxed{\texttt{\textbackslash put}(x,y)\texttt{\{\textbackslash vector}(x_1,y_1)\texttt{\{}\mathit{longueur}\texttt{\}\}}}$

Pour les flèches, les composants du vecteur de direction sont encore plus restreints que pour les segments, aux entiers

$$-4, -3, \ldots, 3, 4.$$

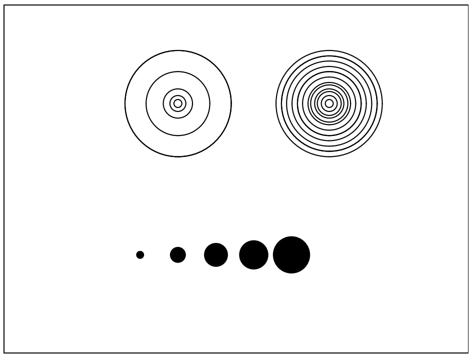
qui doivent aussi être premiers entre eux. Remarquez l'effet de la commande `\thicklines` sur les flèches qui pointent vers le coin supérieur gauche.

5.2.4 Cercles

```
\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}

  \put(40,30){\circle{1}}
  \put(40,30){\circle{2}}
  \put(40,30){\circle{3}}
  \put(40,30){\circle{4}}
  \put(40,30){\circle{5}}
  \put(40,30){\circle{6}}
  \put(40,30){\circle{7}}
  \put(40,30){\circle{8}}
  \put(40,30){\circle{9}}
  \put(40,30){\circle{10}}
  \put(40,30){\circle{11}}
  \put(40,30){\circle{12}}
  \put(40,30){\circle{13}}
  \put(40,30){\circle{14}}

  \put(15,10){\circle*{1}}
  \put(20,10){\circle*{2}}
  \put(25,10){\circle*{3}}
  \put(30,10){\circle*{4}}
  \put(35,10){\circle*{5}}
\end{picture}
```



La commande

`\put(x, y){\circle{diamètre}}`

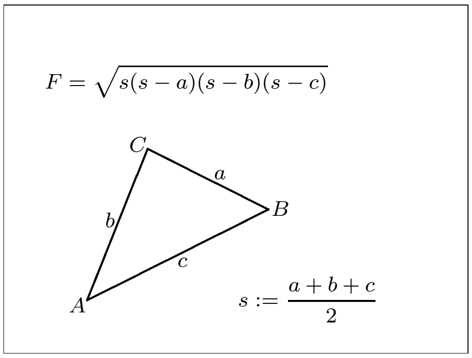
dessine un cercle de centre (x, y) et de diamètre *diamètre* (pas le rayon). L'environnement `picture` admet seulement des cercles de moins de 14 mm, et même en-dessous de cette limite tous les diamètres ne sont pas autorisés. La commande `\circle*` produit quant à elle des disques (i.e. des cercles remplis).

Comme pour les segments, il peut devenir nécessaire de recourir à des extensions supplémentaires comme eepic ou pstricks. Pour une description détaillée de ces extensions, voyez *The LaTeX Graphics Companion* [4].

Il existe aussi une possibilité dans l'environnement `picture`. À condition de ne pas être effrayé par les calculs requis (ou en les laissant à un programme dédié), des cercles et des ellipses de tailles arbitraires peuvent être construits à base de courbes de Bézier, quadratiques. Voyez *Graphics in LaTeX 2ε* [21] pour des exemples et des fichiers source Java.

5.2.5 Texte et formules

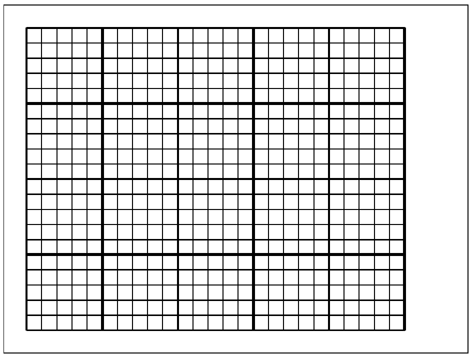
```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){$A$}
  \put(4.05,1.9){$B$}
  \put(1.7,2.95){$C$}
  \put(3.1,2.5){$a$}
  \put(1.3,1.7){$b$}
  \put(2.5,1.05){$c$}
  \put(0.3,4){$F=
    \sqrt{s(s-a)(s-b)(s-c)}$}
  \put(3.5,0.4){$\displaystyle
    s:=\frac{a+b+c}{2}$}
\end{picture}
```



Comme le montre l'exemple ci-dessus, la commande `\put` permet d'insérer du texte et des formules dans un environnement `picture` comme à l'accoutumée.

5.2.6 Les commandes `\multiput` et `\linethickness`

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){26}%
    {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
    {\line(1,0){25}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){6}%
    {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
    {\line(1,0){25}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){3}%
    {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
    {\line(1,0){25}}
\end{picture}
```



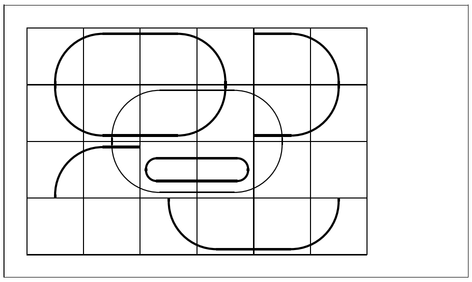
La commande

$\boxed{\texttt{\textbackslash multiput}(x,y)(\Delta x,\Delta y)\{n\}\{\mathit{objet}\}}$

possède 4 paramètres : le point de départ, le vecteur de translation d'un objet à l'autre, le nombre d'objets et l'objet à dessiner. La commande `\linethickness` s'applique aux segments horizontaux et verticaux mais pas aux segments obliques ni aux cercles. Elle s'applique cependant aux courbes de Bézier.

5.2.7 Ovales. Les commandes \thinlines et \thicklines

```
\setlength{\unitlength}{0.75cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \thinlines
  \put(3,2){\oval(3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[tl]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}
```



La commande

$\boxed{\texttt{\textbackslash put(}x,y\texttt{)\{\textbackslash oval(}w,h\texttt{)\}}}$

or

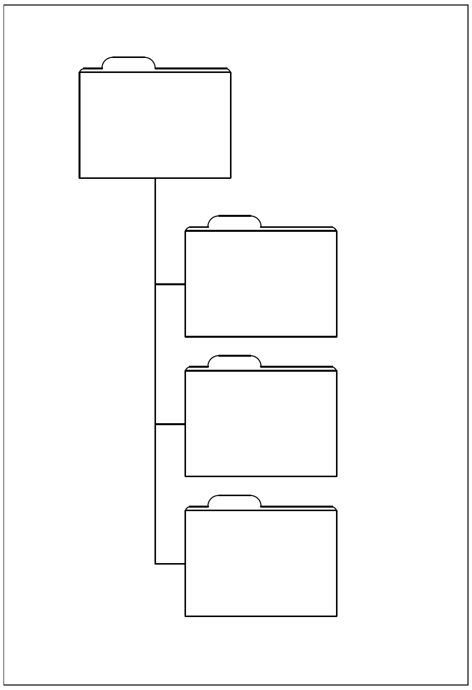
$\boxed{\texttt{\textbackslash put(}x,y\texttt{)\{\textbackslash oval(}w,h\texttt{)[}\textit{position}\texttt{]\}}}$

produit un ovale centré en (x, y), de largeur w et d'hauteur h. Les paramètres optionnels de *position* `b`, `t`, `l` et `r` se rapportent à "haut", "bas","gauche" et "droite" respectivement. Ils peuvent être combinés comme l'illustre l'exemple plus haut.

L'épaisseur de trait peut être contrôlées par deux sortes de commandes : `\linethickness{`*longueur*`}` d'un côté, ou `\thinlines` et `\thicklines` de l'autre. Alors que `\linethickness{`*longueur*`}` ne s'applique qu'aux lignes horizontales, verticales, ou courbes de Bézier, `\thinlines` et `\thicklines` s'appliquent aux segments obliques ainsi qu'aux cercles et aux ovales.

5.2.8 Usage multiple d'images prédéfinies

```
\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}
\savebox{\foldera}
  (40,32)[bl]{% definition
  \multiput(0,0)(0,28){2}
    {\line(1,0){40}}
  \multiput(0,0)(40,0){2}
    {\line(0,1){28}}
  \put(1,28){\oval(2,2)[tl]}
  \put(1,29){\line(1,0){5}}
  \put(9,29){\oval(6,6)[tl]}
  \put(9,32){\line(1,0){8}}
  \put(17,29){\oval(6,6)[tr]}
  \put(20,29){\line(1,0){19}}
  \put(39,28){\oval(2,2)[tr]}
}
\newsavebox{\folderb}
\savebox{\folderb}
  (40,32)[l]{% definition
  \put(0,14){\line(1,0){8}}
  \put(8,0){\usebox{\foldera}}
}
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
  {\usebox{\folderb}}
\end{picture}
```



Une boîte d'image peut être *déclarée* par la commande

`\newsavebox{`*nom*`}`

puis *définie* par

`\savebox{`*nom*`}(`*largeur,hauteur*`)[`*position*`]{`*contenu*`}`

et enfin dessinée arbitrairement souvent via

`\put(`x, y`){\usebox{`*nom*`}}`

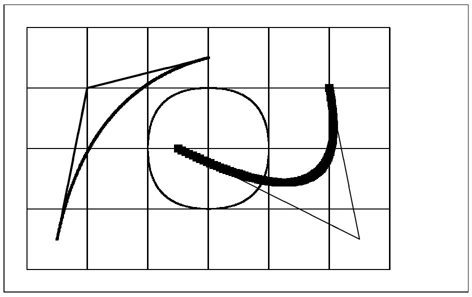
Le paramètre optionnel de *position* a pour effet de définir le "point d'ancrage" de l'image emboîtée. Dans l'exemple il est défini comme `bl` qui met le point d'ancrage dans le coin inférieur gauche de l'image emboîtée. Les autres positions possibles sont `top` (en haut) et `right` (à droite).

Le paramètre *nom* se rapporte à un registre de stockage LaTeX: il s'agit donc d'une commande (d'où l'usage de contre-obliques dans l'exemple). Les images emboîtées peuvent être imbriquées : `\foldera` est utilisée à l'intérieur de la définition de `\folderb` dans l'exemple.

La commande `\oval` a dû être utilisée car la commande `\line` ne fonctionne pas avec des segments dont la longueur est inférieure à 3 mm.

5.2.9 Courbes de Bézier

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qbezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){.5}}
  \linethickness{1mm}
  \qbezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qbezier(4,2)(4,3)(3,3)
  \qbezier(3,3)(2,3)(2,2)
  \qbezier(2,2)(2,1)(3,1)
  \qbezier(3,1)(4,1)(4,2)
\end{picture}
```



Cet exemple illustre l'inadéquation du découpage d'un cercle en 4 courbes de Bézier. Au moins 8 d'entre elles sont requises. La figure montre à nouveau l'effet de la commande `\linethickness` sur les lignes horizontales et verticales, ainsi que des commandes `\thinlines` et `\thicklines` sur les segments obliques. Elle montre enfin l'effet des deux sortes de commandes sur les courbes de Bézier, chaque commande supplantant toutes les précédentes.

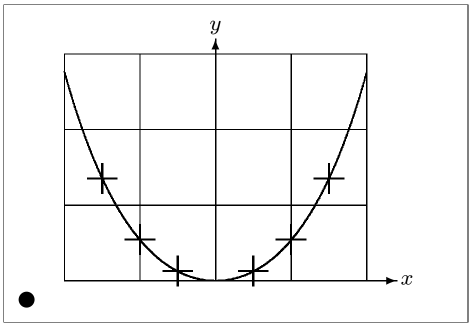
Soient $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ les points de début et fin et m_1, m_2 leurs pentes respectives d'une courbe de Bézier. Le point de contrôle intermédiaire est donné par les équations suivantes :

$$\left\{ \begin{aligned} rclx &= \frac{m_2x_2 - m_1x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y &= y_i + m_i(x - x_i) \qquad (i = 1,\, 2). \end{aligned} \right. \tag{5.1}$$

Référez-vous à *Graphics in* LaTeX 2ε [21] pour un programme Java qui génère les lignes de commandes `\qbezier` nécessaires.

5.2.10 Caténaire

```
\setlength{\unitlength}{1cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){$x$}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){$y$}}
\qbezier(0.0,0.0)(1.2384,0.0)
  (2.0,2.7622)
\qbezier(0.0,0.0)(-1.2384,0.0)
  (-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
  {\line(0,1){3}}
\multiput(-2,0)(0,1){4}
  {\line(1,0){4}}
\linethickness{.2mm}
\put( .3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}
```



Dans cette figure, chaque moitié symmétrique du caténaire $y = \cosh x - 1$ est approchée par une courbe de Bézier. La moitié droite s'achève au point (2, 2.7622) avec une pente $m = 3.6269$. À l'aide de l'équation (5.1) nous pouvons calculer les points de contrôle intermédiaires. Ils s'avèrent être (1.2384, 0) et (−1.2384, 0). Les croix indiquent les points du caténaire *réel*. L'erreur est négligeable, de moins d'un pourcent.

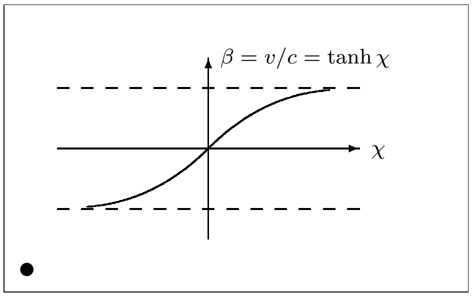
Cet exemple signale l'usage du paramètre optionel de la commande `\begin{picture}`. L'image est définie selon des coordonnées "mathématiques" commodes, tandis que par la commande

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

son coin inférieur gauche (marqué par une disque noir) est associé aux coordonnées $(-2.5, -0.25)$.

5.2.11 La rapidité dans la théorie de la relativité restreinte

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){$\chi$}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {$\beta=v/c=\tanh\chi$}
  \qbezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qbezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
  \put(-3,-2){\circle*{0.2}}
\end{picture}
```



Les points de contrôle des deux courbes de Bézier ont été calculées grâce aux formules (5.1). La branche positive est déterminée par $P_1 = (0, 0)$, $m_1 = 1$ and $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$. L'image est à nouveau définie selon des coordonnées mathématiques commodes et le coin inférieur gauche est associé aux coordonnées mathématiques $(-3, -2)$ (le disque noir).

5.3 Les extensions graphiques PGF et TikZ

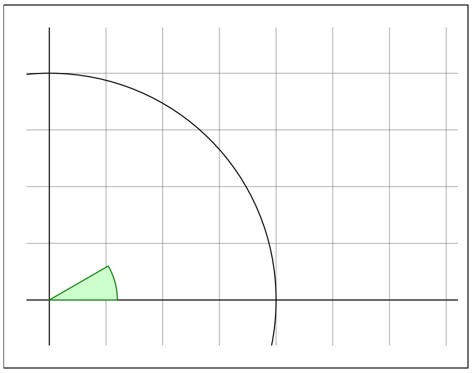
De nos jours, tout système de sortie LaTeX (`dvips`, `pdflatex`, etc.) est capable de fabriquer d'agréables images vectorielles, mais aucun ne le fait avec la même interface. L'extension `pgf` fournit une couche d'abstraction au-dessus de ces interfaces. Elle est fournie avec sa documentation de 500 pages et plus [22]. Nous allons nous contenter d'une vue superficielle de l'utilisation de cette extension dans cette courte section.

L'extension `pgf` est accompagnée par un langage de haut niveau fourni par l'extension `tikz`. TikZ fournit des commandes efficaces pour le dessin d'images directement dans votre document. Utilisez l'environnement `tikzpicture` pour y mettre ces instructions de dessin décrites en TikZ.

Comme mentionné plus haut, il existe un très bon manuel pour `pgf` et ses dérivés. Plutôt donc que montrer comment ces extensions fonctionnent, nous allons juste montrer quelques exemples de ce qui peut être produit afin d'offrir une première impression de l'utilisation de cet outil.

Tout d'abord un diagramme sans signification spéciale :

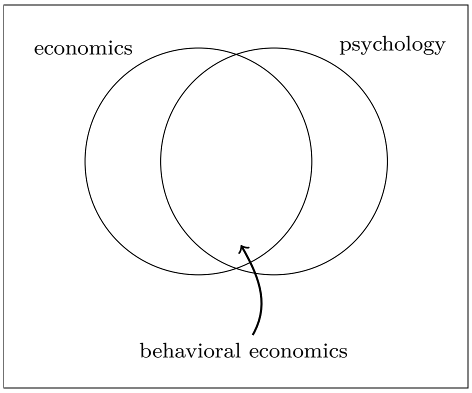
```
\begin{tikzpicture}[scale=3]
  \clip (-0.1,-0.2)
     rectangle (1.8,1.2);
  \draw[step=.25cm,gray,very thin]
       (-1.4,-1.4) grid (3.4,3.4);
  \draw (-1.5,0) -- (2.5,0);
  \draw (0,-1.5) -- (0,1.5);
  \draw (0,0) circle (1cm);
  \filldraw[fill=green!20!white,
            draw=green!50!black]
    (0,0) -- (3mm,0mm)
         arc (0:30:3mm) -- cycle;
\end{tikzpicture}
```



Remarquez l'usage du point-virgule "`;`". Il sépare les commandes individuelles.

Un simple diagramme de Venn.

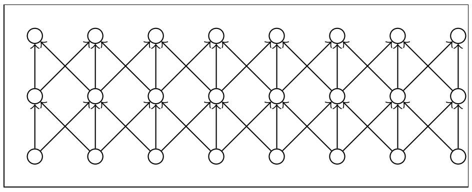
```
\shorthandoff{:}
\begin{tikzpicture}
  \node[circle,draw,
        minimum size=3cm,
        label=120:{economics}]
         at (0,0) {};
  \node[circle,draw,
        minimum size=3cm,
        label=60:{psychology}]
         at (1,0) {};
  \node (i) at (0.5,-1) {};
  \node at (0.6,-2.5)
    {behavioral economics}
    edge[->,thick,
         out=60,in=-60] (i);
\end{tikzpicture}
```



Si vous utilisez `tikz` avec `babel`, il se peut que certains caractères utilisés par le langage TikZ soient modifiés par `babel`, ce qui conduit à des erreurs étranges. Pour éviter ce problème, ajoutez la commande `\shorthandoff` à votre code.

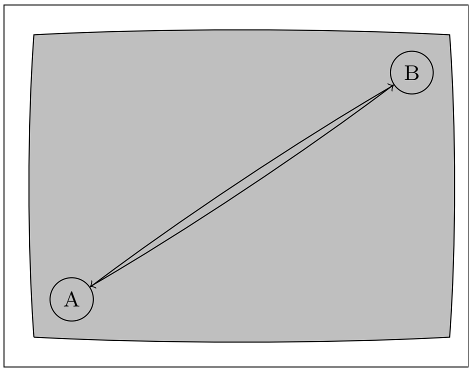
Remarquez les boucles "pour"(`foreach`) dans l'example suivant :

```
\begin{tikzpicture}[scale=0.8]
  \tikzstyle{v}=[circle, minimum size=2mm,inner sep=0pt,draw]
  \foreach \i in {1,...,8}
    \foreach \j in {1,...,3}
      \node[v]
        (G-\i-\j) at (\i,\j) {};
  \foreach \i in {1,...,8}
    \foreach \j/\o in {1/2,2/3}
      \draw[->]
        (G-\i-\j) -- (G-\i-\o);
  \foreach \i/\n in
    {1/2,2/3,3/4,4/5,5/6,6/7,7/8}
    \foreach \j/\o in {1/2,2/3} {
       \draw[->] (G-\i-\j) -- (G-\n-\o);
       \draw[->] (G-\n-\j) -- (G-\i-\o);
    }
\end{tikzpicture}
```

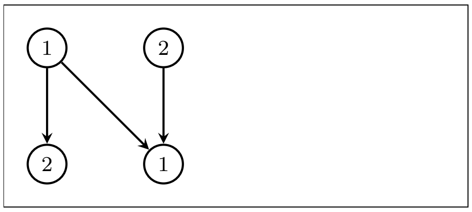


Avec `\usetikzlibrary` dans le préambule, vous pouvez activer de nombreuses fonctionnalités pour le dessin de formes spéciales. Par exemple, la bibliothèque `decorations.pathmorphing` permet d'obtenir des boîtes légérement tordues.

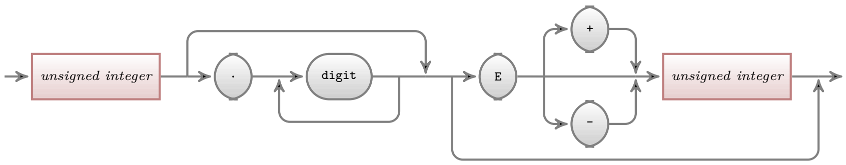
```
\usetikzlibrary{%
  decorations.pathmorphing}
\begin{tikzpicture}[
     decoration={bent,aspect=.3}]
 \draw [decorate,fill=lightgray]
        (0,0) rectangle (5.5,4);
 \node[circle,draw]
        (A) at (.5,.5) {A};
 \node[circle,draw]
        (B) at (5,3.5) {B};
 \draw[->,decorate] (A) -- (B);
 \draw[->,decorate] (B) -- (A);
\end{tikzpicture}
```



```
\usetikzlibrary{positioning}
\begin{tikzpicture}[xscale=6,
     yscale=8,>=stealth]
  \tikzstyle{v}=[circle,
     minimum size=1mm,draw,thick]
  \node[v] (a) {$1$};
  \node[v] (b) [right=of a] {$2$};
  \node[v] (c) [below=of a] {$2$};
  \node[v] (d) [below=of b] {$1$};
  \draw[thick,->]
        (a) to node {} (c);
  \draw[thick,->]
        (a) to node {} (d);
  \draw[thick,->]
        (b) to node {} (d);
\end{tikzpicture}
```



Vous pouvez même dessiner des diagrammes syntaxiques comme s'ils provenaient directement d'un livre sur la programmation Pascal. Le code est plus complexe que l'exemple du dessus, aussi me contenterai-je de ne vous montrer que son résultat. Si vous lisez la documentation de pgf, vous y trouverez un tutoriel pour dessiner ces mêmes diagrammes.



Il y a bien plus : vous pouvez dessiner des graphes de données numériques, de fonctions, etc à l'aide de l'extension pgfplot. Elle fournit tout ce dont vous avez besoin pour dessiner des graphes. Elle peut même appeler la commande externe `gnuplot` pour évaluer les fonctions faisant partie du graphe.

Pour encore plus d'inspiration, visitez l'ébahissant site de Kjell Magne Fauske `http://www.texample.net/tikz/`. Il contient un corpus toujours grandissant de beaux graphiques et de code LaTeX.

Chapter 6

Personnalisation de LaTeX

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

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

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

6.1 Vos propres commandes, environnements et extensions

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

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

`\dum`

Dans cet exemple, nous utilisons à la fois un nouvel environnement appelé `lscommand` qui est responsable du tracé du cadre et une nouvelle commande appelée `\ci` qui compose le nom de la commande et qui produit l'entrée correspondante dans l'index. Vous pouvez le vérifier en cherchant la commande

`\dum` dans l'index à la fin de ce document ; vous y trouverez une entrée pointant vers chaque page où la commande dum est mentionnée.

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

6.1.1 Nouvelles commandes

Pour ajouter de nouvelles commandes, utilisez la commande :

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

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

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

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

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

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

```
\newcommand{\uxil}[2]
    {Une \emph{#1}
     introduction à #2}
% dans le document :
\begin{itemize}
  \item \uxil{courte}{\LaTeXe}
  \item \uxil{longue}{Word}
\end{itemize}
```

– Une *courte* introduction à LaTeX 2ε
– Une *longue* introduction à Word

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

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

LaTeX supprime en général les espaces qui suivent une commande (voir page 5).

6.1.2 Nouveaux environnements

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

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

Il est ici aussi possible d'utiliser `\newenvironment` avec un paramètre optionnel. Le contenu de l'argument *avant* est exécuté avant que le contenu de l'environnement ne soit traité. Le contenu de l'argument *après* est traité lorsque l'on rencontre la commande `\end{`*nom*`}`.

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

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

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

■ Mes chers sujets, ... ■

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

Les commandes utilisées dans l'exemple ci-dessus seront présentées plus loin. Pour la commande `\rule`, voir page 137, pour `\stretch`, voir page 131 enfin, pour plus d'informations sur `\hspace`, voir page 131.

6.1.3 Espaces surnuméraires

La création d'un nouvel environnement est souvent accompagnée du problème récurrent d'espaces surnuméraires qui peuvent avoir des effets indésirables. Par exemple lorsque vous voulez créer un environnement de titre qui supprime sa propre indentation ainsi que celle du paragraphe qui le suit immédiatement. La commande `\ignorespaces` dans le bloc de début de l'environnement lui fera ignorer tout espace situé après l'exécution du bloc de début. Le bloc de fin est plus délicat car un processus spécifique se déroule à la fin d'un environnement. Avec la commande `\ignorespacesafterend` LaTeX ajoutera un `\ignorespaces` qui ignorera les espaces après que le processus spécifique se soit déroulé.

```
\newenvironment{simple}%
 {\noindent}%
 {\par\noindent}

\begin{simple}
Voyez l'espace\\à gauche.
\end{simple}
Ici\\aussi.
```

 Voyez l'espace
à gauche.

 Ici
aussi.

```
\newenvironment{correct}%
 {\noindent\ignorespaces}%
 {\par\noindent%
   \ignorespacesafterend}

\begin{correct}
Plus d'espace\\à gauche.
\end{correct}
Ici\\non plus.
```

Plus d'espace
à gauche.

Ici
non plus.

6.1.4 LaTeX en ligne de commande

Si vous travaillez sur un système de type Unix, il est probable que vous utilisiez des Makefiles pour compiler vos projets LaTeX. Dans ce contexte il pourrait être intéressant de pouvoir produire des versions sensiblement différentes du même document par le simple changement de paramètres de ligne de commande lors de l'invocation de LaTeX. Si vous ajoutez la structure suivante à votre document :

```
\usepackage{ifthen}
\ifthenelse{\equal{\noiretblanc}{true}}{
  % mode noir et blanc; faire ceci..
}{
  % mode couleur; faire cela..
}
```

Alors vous pouvez invoquer LaTeX de la manière suivante:

```
latex '\newcommand{\noiretblanc}{true}\input{document.tex}'
```

D'abord la commande `\noiretblanc` est définie, puis le document est effectivement lu. En modifiant `\noiretblanc` à `false` ce serait la version couleur du document qui serait produite.

6.1.5 Votre propre extension

Si vous définissez beaucoup de nouveaux environnements et de nouvelles commandes, le préambule de votre document va se rallonger dangereusement. Il peut alors devenir intéressant de créer une extension contenant toutes ces

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

Figure 6.1: Exemple d'extension

nouvelles définitions. Avec la commande `\usepackage` vous pourrez rendre disponible votre extension dans votre document.

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

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

`\ProvidesPackage` indique à LaTeX 2ε le nom de l'extension afin notamment de lui permettre de produire des messages d'erreur significatifs. La figure 6.1 montre un exemple d'extension simple qui reprend les commandes définies dans les exemples précédents.

6.2 Polices et tailles des caractères

6.2.1 Commandes de changement de police

LaTeX choisit la police de caractères et sa taille en fonction de la structure logique du document (sections, notes de bas de page, etc.). Dans certains cas, on voudrait pouvoir changer la taille de la police à la main. Pour cela, utilisez les commandes listées dans les tableaux 6.1 et 6.2. La taille exacte de chaque police est un choix qui dépend de la classe de document et de ses options. La table 6.3 donne les tailles absolues en points pour les commandes présentes dans les classes de document standard.

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

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

Une caractéristique importante de LaTeX 2ε est que les différents attributs d'une police sont indépendants[1]. Cela signifie que vous pouvez exécuter des

1. Ou presque. Par exemple, comme l'italique et les petites capitales relèvent d'un

commandes de changement de taille ou même de changement de police tout en conservant l'attribut gras ou italique.

En mode *mathématique*, vous pouvez utiliser les commandes de changement de police pour quitter provisoirement le mode mathématique et saisir du texte normal. Pour changer les attributs de la police en mode mathématique, il existe un jeu de commandes spéciales. Reportez-vous au tableau 6.4.

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

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

Il aime les grands et les petits caractères.

Les commandes de changement de taille modifient également l'interligne, mais seulement si le paragraphe se termine dans la portée de la commande

même attribut pour LaTeX, il lui est impossible de concevoir des petites capitales italiques. (NdT)

Table 6.1: Polices

`\textrm{...}`	romain	`\texttt{...}`	`machine à écrire`
`\textsf{...}`	sans serif		
`\textmd{...}`	moyen	`\textbf{...}`	**gras**
`\textup{...}`	droit	`\textit{...}`	*italique*
`\textsl{...}`	*penché*	`\textsc{...}`	PETITES CAPITALES
`\emph{...}`	*en évidence*	`\textnormal{...}`	par défaut

Table 6.2: 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		

Table 6.3: Tailles en points dans les classes standard

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

Table 6.4: Polices mathématiques

`\mathrm{...}`	Police romaine
`\mathbf{...}`	**Police grasse**
`\mathsf{...}`	Police sans serif
`\mathtt{...}`	`Police typewriter`
`\mathit{...}`	*Police italique*
`\mathcal{...}`	$\mathcal{P}$
`\mathnormal{...}`	*Police normale*

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 [2] :

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

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

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

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

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

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

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

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

6.2.2 Attention danger

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

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

Défense d'**entrer**. Cette pièce contient des **machines** d'origine inconnue.

2. \par est équivalent à une ligne vide

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

-Remarquez la différence entre demander à LaTeX de *mettre en -valeur* un mot et lui demander de changer de *police de -caractères*. La commande `\emph` prend en compte le contexte alors que les commandes de police sont absolues.

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

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

6.2.3 Un conseil

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

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

6.3 Espacement

6.3.1 Entre les lignes

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

`\linespread{`*facteur*`}`

dans le préambule de votre document. Utilisez `\linespread{1.3}` pour un interligne "un et demi" et `\linespread{1.6}` pour un "double" interligne. L'interligne par défaut est 1.

Notez que l'effet de la commande `\linespread` est drastique et inapproprié pour la publication. Si vous avez une bonne raison de changer l'interligne, vous pouvez utiliser la commande :

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
          {1.5\baselineskip}
Ce paragraphe est formatté avec
un interligne fixé à 1,5 de ce
qu'il était avant. Remarquez la
commande par à la fin du
paragraphe.\par}

Ce paragraphe a un objet précis
et montre qu'après l'accolade
fermante tout redevient normal.
```

Ce paragraphe est formatté avec un interligne fixé à 1,5 de ce qu'il était avant. Remarquez la commande par à la fin du paragraphe.

Ce paragraphe a un objet précis et montre qu'après l'accolade fermante tout redevient normal.

6.3.2 Mise en page d'un paragraphe

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

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

dans le préambule, vous supprimez le retrait des débuts de paragraphe (1re définition) et vous augmentez l'espace entre deux paragraphes (2^{e} définition).

Les arguments *plus* et *minus* de la deuxième définition indiquent à TeX de quelle taille il est autorisé à étendre et rétrécir l'espace entre paragraphes, si cela lui est nécessaire pour faire tenir un paragraphe dans une même page.

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

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

```
\indent
```

au début du paragraphe[3]. Bien sûr cela ne marche que si `\parindent` n'est pas nul.

Pour créer un paragraphe sans indentation, utilisez :

```
\noindent
```

en tête du paragraphe.

3. Pour indenter systématiquement le premier paragraphe après le titre d'une section, utilisez[4] l'extension indentfirst de l'ensemble tools.

4. Ou bien, si vous écrivez en français, ne faites rien : babel s'en est chargé pour vous ! (NdT)

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

Il est possible de commencer un paragraphe par une lettrine en utilisant l'extension lettrine [5] :

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

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

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

6.3.3 Espacement horizontal

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

`\hspace{`*longueur*`}`

Si une telle espace doit être conservée, même lorsqu'elle tombe en début ou en fin de ligne, utilisez `\hspace*`. Dans le cas le plus simple, *longueur* est simplement un nombre suivi d'une unité. Les unités les plus importantes sont listées dans le tableau 6.5.

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

Ceci   est une espace de 1.5 cm.

La commande :

`\stretch{`*n*`}`

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

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

x   x   x

Lors que l'espacement horizontal est utilisé en conjonction avec du texte, il est préférable de faire que l'espace soit ajusté en fonction de la taille de la police courante. Ceci peut être réalisé en utilisant les unités relatives `em` et `ex`:

```
{\Large{}plus gr\hspace{1em}and}\\
{\tiny{}minuscu\hspace{1em}le}
```

plus gr and
minuscu le

5. `CTAN:/macros/latex/contrib/supported/lettrine/`

Table 6.5: Unités TeX

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

a. *Inch* en anglais. (NdT).

6.3.4 Espacement vertical

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

`\vspace{`*longueur*`}`

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

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

```
Du texte \dots

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

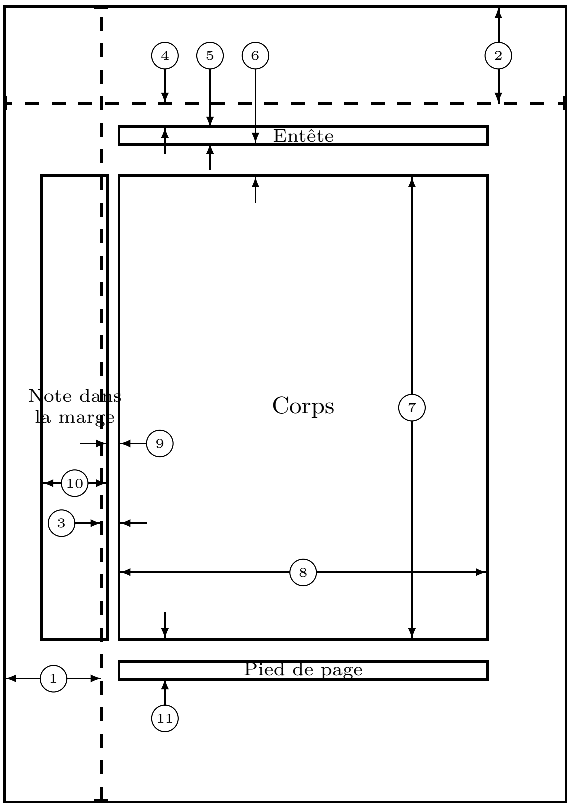
De l'espace supplémentaire entre deux lignes du *même* paragraphe ou à l'intérieur d'une table peut être obtenu par la commande :

`\\[`*longueur*`]`

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

6.4 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



1	`un pouce + \hoffset`	2	`un pouce + \voffset`
3	`\oddsidemargin = -14pt` `or \evensidemargin`	4	`\topmargin = 18pt`
5	`\headheight = 12pt`	6	`\headsep = 25pt`
7	`\textheight = 348pt`	8	`\textwidth = 276pt`
9	`\marginparsep = 10pt`	10	`\marginparwidth = 48pt`
11	`\footskip = 30pt`		`\marginparpush = 5pt (non affiché)`
	`\hoffset = 0pt`		`\voffset = 0pt`
	`\paperwidth = 421pt`		`\paperheight = 597pt`

Figure 6.2: Paramètres de disposition de page pour ce document. Utilisez l'extension layout pour afficher la disposition de page de votre document

et vouloir les modifier. La figure 6.2 montre tous les paramètres qui peuvent être modifiés. Cette figure a été réalisée avec l'extension layout de l'ensemble tools [6].

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

Sans doute, comparé avec une page standard produite avec MS Word, une page de LaTeX à l'air horriblement étroite. Mais regardez votre livre préféré [7] et comptez le nombre de caractères sur une ligne normale. Vous verrez qu'il n'y a guère plus de soixante-six caractères par ligne. L'expérience montre qu'un texte devient moins lisible si le nombre de caractères par ligne dépasse cette valeur, cela parce qu'il devient plus difficile pour les yeux de passer de la fin d'une ligne au début de la ligne suivante. Ceci explique aussi que les journaux utilisent plusieurs colonnes.

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

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

La première commande affecte une valeur fixe au paramètre spécifié :

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

La deuxième commande ajoute une longueur à ce paramètre :

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

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

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

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

6.5 Jouons un peu avec les dimensions

Autant que possible nous évitons d'utiliser des dimensions absolues dans des documents LaTeX. Nous essayons plutôt de les définir relativement aux

6. `macros/latex/required/tools`
7. Un vrai livre, imprimé par un grand éditeur...

dimensions d'autres éléments de la page. La largeur d'une figure sera ainsi `\textwidth` afin de lui faire occuper toute la largeur de la page.

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

```
\settoheight{variable}{texte}
\settodepth{variable}{texte}
\settowidth{variable}{texte}
```

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

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

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

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

Où : a, b sont les cotés adjacents à l'angle droit d'un triangle rectangle,

c est l'hypothénuse du triangle,

d n'est pas utilisé ici. Étonnant non ?

6.6 Boîtes

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

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

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

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

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

on peut également utiliser un environnement :

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

le paramètre `pos` peut être l'une des lettres `c`, `t` ou `b` pour contrôler l'alignement vertical de la boîte par rapport à la base du texte précédent. `largeur` est une dimension indiquant la largeur de la boîte. La différence majeure entre `minipage` et `\parbox` est qu'il est possible d'utiliser quasiment n'importe quelle commande ou environnement dans `minipage`, alors que ce n'est pas le cas [8] avec `\parbox`.

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

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

Le paramètre `largeur` définit la largeur de la boîte vue de l'extérieur [9]. En plus des expressions exprimant une longueur vous pouvez également utiliser `\width`, `\height`, `\depth` et `\totalheight` à l'intérieur du paramètre `largeur`. Leurs valeurs sont obtenues à partir des dimensions réelles du `texte`. Le paramètre `pos` est une lettre parmi `c` (**c**enter) pour centrer le texte, `l` (flush**l**eft) pour l'aligner à gauche, `r` (flush**r**ight) pour l'aligner à droite, ou `s` (**s**pread) pour le répartir horizontalement dans la boîte.

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

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

---

8. En fait, ce sont essentiellement les environnements et commandes de type *verbatim* qui posent problème. (NdT)

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

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

c e n t r é

é t i r é

Whoua le cadre !

Raté, je suis trop large

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

Maintenant que nous savons contrôler l'alignement horizontal, la suite logique est de voir comment gérer l'alignement vertical [10]. Pas de problème avec LaTeX. La commande :

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

permet de définir les propriétés verticales d'une boîte. La *profondeur* correspond à une extension *sous* la ligne de base du texte, la *hauteur* à une extension *au-dessus* de cette ligne. Vous pouvez utiliser `\width`, `\height`, `\depth` et `\totalheight` dans les trois premiers paramètres afin d'agir en fonction de la taille du texte contenu dans la boîte.

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

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

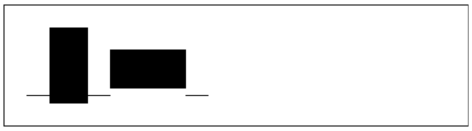
6.7 Filets

Quelques pages plus haut vous avez peut-être remarqué la commande :

`\rule[`*élévation*`]{`*largeur*`}{`*hauteur*`}`

En utilisation normale, elle produit une simple boîte noire.

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



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

C'est utile pour produire des lignes horizontales et verticales. La ligne horizontale sur la page de titre par exemple a été tracée à l'aide d'une commande `\rule`, par exemple.

Fin.

Appendix A

Installation de LaTeX

Knuth a diffusé les sources de TeX à une époque où personne ne savait ce qu'étaient les concepts de logiciel libre ou à source ouvert. La licence fournie avec TeX vous laisse faire tout ce que vous voulez avec les sources, mais vous ne pouvez appeler le résultat TeX que si le programme passe une série de tests fournis aussi par Knuth. Ceci a mené à une situation où il existe une implantation libre de TeX pour presque tous les systèmes d'exploitation existants. Dans ce chapitre nous donnerons quelques astuces sur ce dont vous avez besoin pour faire fonctionner TeX sous Linux, Mac OS X et Windows.

A.1 Ce qu'il faut installer

Pour installer LaTeX sur un ordinateur, les logiciels suivants sont essentiels :

1. Le programme TeX/LaTeX pour compiler vos fichiers sources LaTeX en documents PDF ou DVI ;
2. Un éditeur de texte pour éditer vos fichiers sources LaTeX. Certains programmes permettent même le lancement du programme LaTeX depuis l'éditeur ;
3. Un visualiseur PDF ou DVI pour afficher et imprimer vos documents ;
4. Un programme pour gérer les fichiers PostScript et les images pour inclusion dans vos documents.

Il y a tant de programmes qui rentrent dans les critères définis ci-dessus... Nous discuterons ici seulement de ceux que nous connaissons, apprécions et pour lesquels nous avons quelque expérience.

A.2 TeX sous Mac OS X

A.2.1 Obtenir une distribution TeX

Téléchargez tout simplement MacTeX. Il s'agit d'une distribution précompilée de LaTeX pour OSX. MacTeX met à disposition une installation complète de LaTeX fournie avec des outils supplémentaires. Obtenez MacTex sur `http://www.tug.org/mactex/`.

Si vous utilisez déjà les Macports ou Fink pour installer des logiciels Unix sous OSX, utilisez-les aussi pour LaTeX. Avec Macports la commande sera `port install texlive` et avec Fink `fink install texlive`.

A.2.2 Choisir un éditeur

L'éditeur ouvert le plus populaire pour LaTeX sous Mac semble être TeXshop. Téléchargez-en une copie sur `http://www.uoregon.edu/~koch/texshop`. Il est aussi fourni avec la distribution MacTeX.

Texmaker est un autre bon éditeur. Son avantage, à part d'être un bon éditeur, est sa capacité à fonctionner sous Windows, Mac et Unix/Linux. Visitez `http://www.xm1math.net/texmaker` pour plus d'information. Notez qu'il existe également une version dérivée de Texmaker appelée TexmakerX sur `http://texmakerx.sourceforge.net/` qui y ajoute de nouvelles fonctionnalités.

Les distributions récentes TeXLive contiennent l'éditeur TeXworks `http://texworks.org/`, un éditeur multi-plateformes basé sur les concepts de TeXShop. Puisque TeXworks utilise la bibliothèque Qt, il est disponible sur toute plateforme sur laquelle cette bibliothèque fonctionne (MacOS X, Windows, Linux).

A.2.3 Faites-vous plaisir avec PDFView

Utilisez PDFView pour afficher des fichiers PDF générés par LaTeX, il s'intègre parfaitement avec votre éditeur. PDFView est une application libre qui peut être téléchargée depuis `http://pdfview.sourceforge.net/`. Une fois téléchargé et installé, ouvrez les préférences de PDFView et assurez-vous que l'option *recharger/rafraîchir automatiquement les documents* est activée et que le support PDFSync est positionné sur le réglage “TextMate”.

A.3 TeX sous Windows

A.3.1 Obtenir TeX

En premier lieu, obtenez une copie de l'excellente distribution MiKTeX sur `http://www.miktex.org/`. Elle contient tous les programmes et fichiers de base pour compiler des documents LaTeX. La fonctionnalité la plus sympathique, à mes yeux, est que MiKTeX téléchargera les extensions LaTeX manquantes à la volée et les installera lors de la compilation d'un document. Vous pouvez aussi utiliser la distribution TeXlive pour Windows, Unix et Mac OS pour obtenir un bon environnement LATEX de base `http://www.tug.org/texlive/`.

A.3.2 Un éditeur LATEX

LATEX est un langage de programmation pour documents. TeXnicCenter fait appel à plusieurs concepts du monde de la programmation pour mettre à disposition un environnement LATEX agréable et efficace pour la saisie sous Windows. Obtenez une copie sur `http://http://www.texniccenter.org/`. TeXnicCenter s'intègre bien avec MiKTeX. La version 2.0 de TeXnicCenter prendra en compte l'Unicode et la version alpha récente semble raisonnablement stable.

Un autre très bon choix est l'éditeur fourni par le projet LEd sur `http://www.latexeditor.org`.

Voyez la note sur Texmaker dans la section Mac plus haut pour un troisième choix.

Les distributions TEXLive récentes contiennent l'éditeur TEXworks `http://texworks.org/`. Celui-ci prend en compte l'Unicode et nécessite Windows XP au minimum.

A.3.3 Prévisualisation

Vous utiliserez probablement Yap pour visualiser votre document DVI, puisqu'il est installé avec MikTeX. Pour un document PDF vous pouvez regarder du côté de Sumatra PDF `http://blog.kowalczyk.info/software/sumatrapdf/`. Nous le mentionnons ici parce qu'il permet après réglages d'aller directement d'une position dans le document PDF à la position correspondante dans le document source.

A.3.4 Travailler avec des images

Travailler avec des images de haute qualité dans LATEX est synonyme d'utilisation de PostScript encapsulé (eps) ou de PDF comme format d'image. Le programme qui vous aidera dans cette tâche s'appelle GhostScript. Vous pouvez l'obtenir avec son interface graphique propre GhostView sur `http://www.cs.wisc.edu/~ghost/`.

Si vous utilisez plutôt des formats bitmap (photos et images numérisées), vous pouvez jeter un oeil à Gimp, une alternative libre à PhotoShop, téléchargeable sur `http://gimp-win.sourceforge.net/`.

A.4 TEX sous Linux

Si vous travaillez sous Linux, il est probable que LATEX soit déjà installé sur votre système, ou au moins disponible sur les dépôts que vous avez utilisés lors de son installation. Utilisez votre gestionnaire de paquets pour installer les applications suivantes :

– texlive – l'installation TEX/LATEX de base ;
– emacs (avec auctex) – un éditeur Linux qui s'intègre étroitement avec LATEX via une extension appelée AucTeX ;

– ghostscript – un visualiseur PostScript ;
– xpdf et acrobat – des visualiseurs PDF ;
– imagemagick – un programme libre pour convertir des images bitmap ;
– gimp – un clone libre de PhotoShop ;
– inkscape – un clone libre d'Illustrator/Corel Draw (dessin vectoriel).

Si vous cherchez un environnement graphique d'édition plus proche de ce qui se fait sous Windows, essayez Texmaker ou TeXworks, éditeurs dont il est question dans la section Mac plus haut.

Attention, la plupart des distributions découpent TeX Live en plusieurs paquets. Si vous souhaitez une installation complète, cherchez un paquet appelé `texlive-full` ou bien installez tous les paquets dont le nom contient `texlive` ou `latex`, ainsi que les paquets `cm-super` et `lmodern` s'ils existent.

Bibliography

[1] LAMPORT, Leslie. *LaTeX: A Document Preparation System.* Addison-Wesley, 1994. 2e édition.
ISBN 0-201-52983-1.

[2] KNUTH, Donald E. *The TeXbook,* Volume A de *Computers and Typesetting.* Addison-Wesley, 1984. 2e édition.
ISBN 0-201-13448-9.

[3] MITTELBACH, Frank ; GOOSSENS, Michel. Trad. supervisée par Jacques ANDRÉ *le LaTeX Companion, 2e édition.* Pearson Education France, 2005.
ISBN 2-7440-7133-1.

[4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The LaTeX Graphics Companion.* Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.

[5] DESGRAUPES, Bernard. *LaTeX Apprentissage, guide et référence.* Vuibert, 2000.
ISBN 2-7117-8658-7.

[6] 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.

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

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

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

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

[11] SCHÖPF, Rainer ; RAICHLE, Bernd et ROWLEY Chris. *A New Implementation of LaTeX's verbatim Environments.* Distribué avec l'ensemble « tools » dans `verbatim.dtx`.

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

[13] Gaulle, Bernard. *Notice d'utilisation du style french multilingue*. Disponible avec l'extension `french` sur `http://frenchpro.free.fr/`.

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

[15] Vladimir Volovich, Werner Lemberg and LaTeX3 Project Team. *Cyrillic languages support in LaTeX*. Fourni avec la distribution LaTeX 2ε sous la forme du fichier `cyrguide.tex`.

[16] Graham Williams. *The TeX Catalogue* is a very complete listing of many TeX and LaTeX related packages. Available online from `CTAN://help/Catalogue/catalogue.html`

[17] Reckdahl, Keith. *Using EPS Graphics in LaTeX 2ε Documents* qui explique tout ce que vous avez toujours voulu savoir et même plus sur les fichiers PostScript et leur utilisation avec LaTeX. Disponible en ligne sur `CTAN://info/epslatex.ps`

[18] Kristoffer H. Rose. *XY-pic User's Guide*. Téléchargeable depuis le CTAN avec la distribution XY-pic

[19] John D. Hobby. *A User's Manual for METAPOST*. Téléchargeable depuis `http://cm.bell-labs.com/who/hobby/`

[20] Alan Hoenig. *TeX Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)

[21] Urs Oswald. *Graphics in LaTeX 2ε*, contient des fichiers source Java pour générer des cercles et des ellipses arbitraires dans l'environnement `picture`, et *METAPOST - A Tutorial*. Les deux sont téléchargeables depuis `http://www.ursoswald.ch`

[22] Till Tantau. *TikZ&PGF Manual*. Download from `CTAN://graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf`

[23] François Charette. *Polyglossia: A Babel Replacement for XeLaTeX*. Comes with the TeXLive distribution as `polyglossia.pdf`. (Type `texdoc polyglossia` on the command line.)

[24] François Charette. *An ArabTeX-like interface for typesetting languages in Arabic script with XeLaTeX*. Comes with the TeXLive distribution as `arabxetex.pdf`. (Type `texdoc arabxetex` on the command line.)

[25] Will Robertson and Khaled Hosny. *The `fontspec` package*. Comes with the TeXLive distribution as `fontspec.pdf`. (Type `texdoc fontspec` on the command line.)

[26] Apostolos Syropoulos. *The `xgreek` package*. Comes with the TeXLive distribution as `xgreek.pdf`. (Type `texdoc xgreek` on the command line.)

[27] Vafa Khalighi. *The `bidi` package*. Comes with the TeXLive distribution as `bidi.pdf`. (Type `texdoc bidi` on the command line.

[28] Vafa Khalighi. *The `XePersian` package*. Comes with the TEXLive distribution as `xepersian-doc.pdf`. (Type `texdoc xepersian` on the command line.

[29] Wenchang Sun. *The `xeCJK` package*. Comes with the TEXLive distribution as `xeCJK.pdf`. (Type `texdoc xecjk` on the command line.

Index