LaTeX
Manuel utilisateur simplifié

Chantal Simian

CNRS LTAMT[1]

Remis à jour par

C. Harmide, P.-V. Koseleff, J. Marchand

J. Moulin-Ollagnier, R. Rioboo, M. Rybowicz et B. Salvy

Version 1.3

May 5, 1992

[1]Laboratoire de Thermodynamique Appliquée aux Machines et Thermique - Saint-Cyr l'Ecole

Table des matières

Chapitre 1

Introduction

LaTeX se veut être une forme simplifiée du célèbre traitement de textes mathématiques TeX de D. Knuth. LaTeX ajoute en effet à TeX bon nombre de commandes de haut niveau et des styles prédéfinis pour les documents, ce qui facilite le travail de l'utilisateur. Tous ceux qui désireraient une description complète et précise de cet outil pourront se référer à l'ouvrage de Leslie Lamport[1].

Nous nous contenterons de donner les commandes les plus importantes pour permettre au lecteur de faire ses premiers pas. Il n'est pas question ici de décrire toutes les possibilités et les subtilités de LaTeX.

[1] A Document Preparation System
LaTeX
User's Guide & Reference Manual
Leslie Lamport
Addison-Wesley

Chapitre 2

Utilisation pratique

LaTeX n'est pas un traitement de texte interactif : la première chose à faire est de créer à l'aide de votre éditeur préféré un fichier source rassemblant votre texte et les commandes LaTeX adéquates. La construction du fichier source est l'objet des chapitres suivants.
La façon dont ce fichier source est ensuite compilé en un document de haute qualité, imprimable ou visualisable sur un écran haute résolution, dépend de votre matériel et de votre installation. Par exemple, sous UNIX, le fichier source doit généralement porter le suffixe *tex* pour être traité par la commande :

latex *nom-de-fichier.tex*

Cette commande produit un fichier nommé *nom-de-fichier.dvi* qui pourra être visualisé sur un terminal X-Window, ou que vous pourrez expédier directement à l'imprimante si vous disposez sur votre site du filtre adéquat. La commande de visualisation à l'écran est en général :

xdvi *nom-de-fichier.dvi*

Si vous ne disposez pas d'un filtre adéquat pour envoyer directement à l'imprimante un fichier *dvi*, il est possible de transformer le fichier *dvi* en fichier PostScript, langage connu de la plupart des imprimantes laser modernes. La commande :

dvips *nom-de-fichier.ps*

produit un fichier PostScript *nom-de-fichier.ps*. Encore une fois, tout ceci dépend de votre installation. Consultez votre administrateur système pour plus de détails.

LaTeX est fort prolixe, donc pas de panique. Si, durant la compilation une erreur est détectée, un message vous le signale et un point d'interrogation s'affiche en début de ligne : LaTeX attend un ordre de votre part. Parmi ceux possibles, voici les plus utiles :

? LaTeX vous donne la liste des commandes possibles.

h LaTeX vous donne un diagnostic détaillé de l'erreur et vous suggère une solution.

e LaTeX appelle l'éditeur et positionne le curseur à l'endroit de l'erreur.

q Arrêt du traitement.

Ce ne sont que des indications. Quelques essais vous permettront de comprendre rapidement les diverses solutions qui s'offrent à vous. Vous trouverez en annexe de l'ouvrage de Leslie Lamport une liste des principales erreurs possibles.

Pour ses besoins propres, LaTeX va créer un certain nombre de fichiers auxiliaires. Ainsi, si vous travaillez sur un fichier *essai.tex*, vous obtiendrez, en plus du fichier *essai.dvi* :

essai.aux	qui contient des informations propres à LaTeX,
essai.log	qui contient la copie de ce qui est apparu sur votre écran durant la compilation,
essai.toc	qui contient la table des matières,
essai.lof	qui contient la table des figures.

Chapitre 3

Le fichier source

3.1 Structure d'un fichier source

Un fichier source se compose de deux parties: le *préambule* et votre document proprement dit, que l'on nommera le *corps* dans la suite.

3.1.1 Le préambule et les styles

Le préambule permet d'indiquer le style de votre document, ainsi que des directives concernant, par exemple, la mise en page, le titre ou les fontes que vous souhaitez utiliser, si elles ne sont pas standard. LaTeX est un langage de programmation et on peut aussi utiliser le préambule pour définir de nouvelles commandes, de nouveaux environnements, ou encore des compteurs. Le préambule commence impérativement par la commande :

`\documentstyle[`*options1,options2,...*`]{`*nom-de-style*`}`

où *nom-de-style* est à choisir parmi *letter, report, article, book*. Les plus utilisés sont les trois derniers mentionnés. Le choix d'un style détermine certaines caractéristiques de votre document, comme par exemple la taille des titres, les commandes de sectionnement autorisées ou encore les indentations. Chaque style admet un certain nombre d'options, qui permettent de préciser par exemple la taille par défaut des caractères.

Sectionnement du texte Un style définit des commandes de sectionnement du texte, qui permettent de structurer hiérarchiquement le document. Selon l'ordre dans la hiérarchie et le style, les tailles des titres et les indentations seront différentes.

On distingue par ordre décroissant :
`\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph`.

Chaque niveau doit être contenu dans celui juste supérieur. Le niveau `\part` est optionnel dans tous les styles. Les commandes `\chapter`, `\section`, `\subection` et `\subsubsection` sont automatiquement numérotées. Toutes ces commandes de sectionnement prennent pour argument le titre de la portion de texte en question. Il faut noter que la commande `\chapter` n'est pas disponible dans le style *article*.

Les options Citons quelques options utiles: les options *11pt* et *12pt* permettent d'obtenir des caractères par défaut plus gros. L'option *twocolumn* permet de présenter le texte sur deux colonnes verticales. L'option *twoside* permet de produire des documents destinés à être imprimés recto-verso.

Titre et page de titre Enfin, le préambule sert aussi à associer un titre, un auteur et une date à un document. La commande `\maketitle` insérée au début du corps aura pour effet de placer ces informations en tête de votre document. Dans les styles *book* et *report*, le titre est placé sur une page indépendante.

Mise en page Pour créer des documents destinés à être imprimés sur du format A4, il est souhaitable de redéfinir la taille de la page. Voir l'exemple ci-dessous. Ainsi, un préambule typique ressemble à:

```
\documentsyle[11pt,twocolumn]{article}

\title{Un titre quelconque}
\author{Peu Importe}
\date{N'importe quand}

\textwidth 16.8cm         % largeur du texte
\textheight 23.5cm        % hauteur du texte
\oddsidemargin -.25in     % marge pour les pages impaires
\evensidemargin -.25in    % marge pour les pages paires
\topskip 0cm
\footskip 0cm
\footheight 0cm
\headheight 0cm
```

L'utilisateur peut aussi définir ses propres options, ou même ses propres styles, ce qui est plus difficile. Voir le manuel de référence pour cela.

3.1.2 Le document

Cette partie du fichier source contient votre texte tapé “au kilomètre”, ainsi que les commandes qui indiquent comment il doit être présenté. Votre texte doit impérativement commencer par un ordre `\begin`{*document*} et se terminer par un `\end`{*document*}. Comme indiqué ci-dessus, la commande `\maketitle` placée après `\begin`{*document*} permet de produire un titre à partir des informations fournies dans le préambule. Ainsi, un corps typique est de la forme :

```
\begin{document}
\maketitle
\section{Introduction}
Du texte........
\section{Th\'{e}ses}
\subsection{Th\'{e}se A}
Du texte.......
\subsection{Th\'{e}se B}
Du texte......
\section{Antith\'{e}ses}
Du texte.......
\section{Synth\'{e}se}
Du texte.......
\end{document}
```

3.2 La composition du fichier source

3.2.1 Les caractères

Le fichier source peut contenir toutes les lettres majuscules et minuscules, les chiffres de 0 à 9 et les seize caractères de ponctuation :

```
. : ; , ! ? ‘ ’ ( ) [ ] - / + @
```

auxquels peuvent s’ajouter les caractères spéciaux :

```
# $ % & - { }
```

qui, réservés aux commandes, doivent être précédés d’un \ lorsque l’on veut les utiliser pour eux-mêmes, et des caractères “très” spéciaux :

~ ^ \

qui peuvent apparaître à l'aide des commandes `\verb` :

```
\~{}  \^{}  $\backslash$
```

Il faut aussi remarquer qu'une suite de blancs (espaces) est comptée pour un seul blanc. Les retours à la ligne ont une signification particulière : deux retours à la ligne consécutifs sont interprétés comme un changement de paragraphe. LaTeX peut générer une multitude d'autres caractères utiles pour le traitement des textes scientifiques. On consultera en particulier le chapitre concernant le mode mathématique.

3.2.2 Les modes

Il existe en LaTeX deux modes de fonctionnement : le mode standard, et le mode mathématique qui permet de taper des formules mathématiques complexes via des commandes adéquates non reconnues en mode standard. Un chapitre de ce manuel est consacré au passage d'un mode à l'autre et aux principales commandes disponibles en mode mathématique. Les sections suivantes ne concernent que le mode standard.

3.2.3 Les accents

La liste qui suit est donnée avec la lettre "e" comme référence, mais la méthode est valable quelle que soit la lettre à accentuer. Ceci ne fonctionne que pour le mode standard. Pour les accents en mode mathématique, consulter le chapitre sur le mode mathématique.

é `\'{e}`	è ``\`{e}``	ê `\^{e}`	ë `\"{e }`
ẽ `\~{e}`	ē `\={e}`	ė `\.{e}`	ĕ `\u{e}`
ě `\v{e}`	e̋ `\H{e}`	e͡e `\t{ee}`	ȩ `\c{e}`
ẹ `\d{e}`	e̱ `\b{e}`		

On remarquera que le i et le j demandent un traitement particulier : tous deux doivent être en plus précédés d'un \ pour ôter le point. Par exemple, on utilisera : `\"{\i}` et non pas `\"{i}`. D'autre part, les accolades autour du "e" ne sont nécessaires que lorsque l'accent n'est pas une lettre (par exemple une cédille).

3.2.4 Quelques symboles particuliers

©	`\copyright`	†	`\dag`	§	`\S`
¶	`\P`	œ	`\oe`	æ	`\ae`

3.2.5 Les commandes

Une commande est :

- soit un caractère spécial tel le ~
- soit un `\` suivi d'un caractère non-alphabétique comme `\@`
- soit un `\` suivi d'un nom de commande.

Les arguments obligatoires sont placés entre `{ }`, et ceux facultatifs entre `[ ]`.

Une déclaration est une commande qui n'imprime rien, mais modifie l'environnement dans lequel sera traité le texte sur lequel elle porte. Celui-ci est inclus dans des `{ }`. La validité d'une déclaration commence avec cette déclaration et s'achève à la fin de l'environnement qui la contient. Cette fin est marquée par une accolade fermante `}` et les règles usuelles du parenthésage s'appliquent. Toutefois, lorsqu'une déclaration concerne une part importante de texte, on préférera employer la forme `\begin`{*nom-de-l'environnement*} au début du texte puis `\end`{*nom-de-l'environnement*} à la fin.

3.2.6 Règles importantes

Pour pouvoir rapidement faire quelques essais, il est important de connaître les quelques règles suivantes :

Pour LaTeX :

- Un mot est une suite de signes non spéciaux, délimitée par un ou plusieurs blancs (il n'en retient qu'un seul). Pour forcer la prise en compte d'un blanc supplémentaire, il faut le faire précéder d'un \.
- Une phrase est une suite de mots terminée par un point.
- Un paragraphe est une suite de phrases terminée par une ou plusieurs lignes blanches (c'est-à-dire au moins deux retours à la ligne consécutifs).

Nous pouvons y ajouter quelques commandes d'usage courant :

- Pour éviter que des mots soient séparés par un passage à la ligne malencontreux, il suffit de les relier par un `~`.
- Pour qu'une portion de texte ne soit pas analysée et donc garde la présentation initiale il faut l'insérer entre un `\begin`{*verbatim*} et un `\end`{*verbatim*}.
- Pour inclure des commentaires dans votre fichier source, il faut les faire précéder d'un `%` LaTeX ignorera alors tous les caractères jusqu'au prochain retour à la ligne.
- En général, LaTeX se charge de la césure des mots. Il peut cependant arriver qu'il faille lui indiquer à quels endroits d'un mot les césures sont autorisées. Par exemple, si LaTeX refuse de couper le mot "effréné", et que ceci provoque un débordement du texte hors de la ligne, on peut autoriser LaTeX à effectuer une césure entre les deux "f" en tapant `ef\-fr\'{e}n\'{e}`.

3.3 Utilisation de plusieurs fichiers

Lorsque vous travaillez sur un texte volumineux, il est plus agréable de le découper et de le répartir en plusieurs fichiers. Vous devez toutefois choisir dans ce cas-là un fichier principal (qui contiendra au moins le préambule et les commandes `\begin`{*document*} et `\end`{*document*}) dans lequel vous indiquerez les différents modules choisis, et sur lequel se fera l'appel de LaTeX.

Deux possibilités existent pour découper votre texte. L'utilisation de la commande `\input`{*nom-de-fichier*} dans le fichier principal permet à LaTeX d'insérer à cet endroit le contenu du fichier *nom-de-fichier.tex*. Vous pouvez avoir plusieurs niveaux d'inclusion.

Une autre commande, assez semblable à `\input` existe, c'est `\include`{*nom-de-fichier*}. Il y a aussi ajout du texte contenu dans le fichier à l'endroit où apparaît cette commande. Toutefois vous pouvez, dans l'entête de votre fichier principal, ajouter une commande `\includeonly`{*nom-de-fichier*} qui sélectionnera les `\include` que vous désirez réellement voir exécuter. C'est particulièrement agréable dans les phases de mise au point. La possibilité pour entrer la commande `\includeonly` d'utiliser `\typein`, qui vous permet d'insérer en interactif une ligne tapée au clavier, est une grande aide à la mise au point, car vous ne touchez jamais à votre fichier principal.

3.3.1 Quelques exemples

```
\documentstyle{report}
\begin{document}
\include{chap1}
```

```
\include{chap2}
\end{document}
```

$\Longrightarrow$ inclusion de chap1, chap2.

```
\documentstyle{report}
\includeonly{chap2}
\begin{document}
\include{chap1}
\include{chap2}
\end{document}
```

$\Longrightarrow$ le traitement ne portera dans ce cas-là que sur chap2

Enfin la forme :

```
\documentstyle{report}
\typein{taper votre `includeonly'}
\begin{document}
\include{chap1}
\include{chap2}
\end{document}
```

$\Longrightarrow$ vous laissera le choix, lors de l'exécution, de travailler sur l'un ou l'autre des chapitres.

Chapitre 4

Les principales commandes

4.1 Le découpage du texte

Voir la section concernant les styles et le préambule.

4.2 Présentation des phrases et des paragraphes

- changement de ligne : `\\`
- changement de page : `\newpage` `\clearpage`
- mise en valeur d'une portion de texte par une indentation des marges et un centrage du texte `\begin{`*quote*`}...\end{`*quote*`}`
- mise en valeur d'une portion de texte par une indentation des marges, centrage du texte et indentation des paragraphes `\begin{`*quotation*`}...\end{`*quotation*`}`
- mise en valeur d'un mot par utilisation des quotes ou des guillemets. Pour des quotes le mot doit être précédé d'une quote ouvrante et suivi d'une fermante. Pour obtenir des guillemets, il suffit de doubler ces signes.
- mise en relief d'un mot par changement de fonte. On utilise la déclaration `\em` et on fait en sorte que la portée de cette déclaration soit exactement le texte à mettre en valeur.

4.3 Les blancs et les espaces

En principe, LaTeX gère automatiquement les blancs horizontaux et verticaux. Il arrive néanmoins fréquemment que l'on souhaite "sauter des lignes" ou laisser un espace blanc horizontal. On peut utiliser pour cela les commandes `\hspace` et `\vspace`. Ces deux commandes prennent un argument qui est la hauteur, ou la longueur à laisser en blanc. Par exemple, `\vspace{`*1.5cm*`}` laisse un espace blanc de hauteur 1.5 centimètre. Dans le même ordre d'idées, la commande `\hspace{\fill}` "remplit" de blanc la ligne à la position courante : ceci peut-être utile, par exemple, pour justifier à droite une partie du texte. La commande `\hspace` (resp. `\vspace`) ne fonctionne pas en fin de ligne (resp. page) : on utilisera dans ce cas `\hspace*` (resp. `\vspace*`).

4.4 Changements de style et de taille des caractères

La fonte utilisée par défaut est la fonte romane (`\rm`). Vous pouvez toutefois utiliser :

- **les caractères gras** `\bf`
- LES PETITES CAPITALES `\sc`
- `le style machine à écrire` `\tt`
- *les caractères italiques* `\it`
- *les caractères pseudo-italiques* `\sl`

La taille des caractères est par défaut de dix points. Vous pouvez cependant facilement la changer pour 11 ou 12 points. C'est un argument à donner dans le choix du style. Vous pouvez aussi désirer ne changer la taille des caractères que d'une partie de votre texte, pour en augmenter ou en réduire la taille. Vous pouvez choisir :

- la plus petite taille `\tiny`
- une taille légèrement plus grande `\scriptsize`
- la taille des footnotes `\footnotesize`
- une petite taille `\small`
- la taille normale `\normalsize`
- une taille légèrement plus grande `\large`
- un peu plus grande `\Large`
- encore plus grande `\LARGE`
- toujours plus grande `\huge`

• la plus grande `\Huge`

Notons qu'en LaTeX, la bonne manière d'insister sur quelque chose est de faire *le texte important*. Il est intéressant de constater que *si l'on est* déja *dans un* `\em`, alors le changement de caractères a lieu à l'envers : si l'on est en pseudo-italique, alors on repasse en roman. Le nom de la commande `\em` est l'abréviation de "emphasize" (insister sur). Pour les perfectionnistes de la typographie, signalons enfin qu'il convient de faire une "correction italique", qui consiste à ajouter avant le changement.
Exemple :
Voici un *espacement* correct, produit par `{\em espacement\/}`.

4.5 Les justifications et le centrage

Vous pouvez justifier une partie de votre texte à gauche (respectivement à droite) ou demander son centrage. Pour cela il suffit d'utiliser les environnements

```
\begin{flushleft}.......\end{flushleft}
\begin{flushright}......\end{flushright}
```

dans les cas de justification et

```
\begin{center}...\end{center}
```

pour le centrage.

4.6 Les annotations

4.6.1 Les renvois en bas de page

La commande `\footnote`{*texte à mettre en renvoi*} permet de créer une note en bas de page et, sauf avis contraire, la numérote automatiquement.

4.6.2 Les annotations en marge

La commande `\marginpar`{*texte de la note*} permet leur création.

4.7 Les environnements

Le concept d'*environnement* est fondamental en LaTeX. Un environnement commence par une commande
`\begin`{*nom-de-l'environnement*}
et se termine par :
`\end`{*nom-de-l'environnement*}
Chaque environnement a son propre mode de mise en forme du texte, d'indentation et de justification. Les environnements permettent d'obtenir toutes sortes d'effets utiles. L'utilisateur peut créer ses propres environnements (voir la section 6.1.2), mais les environnements prédéfinis sont suffisants dans la plupart des cas :

4.7.1 itemize

L'environnement *itemize* permet de produire une suite de paragraphes étiquetés et pourvus d'une marge spéciale. Chaque paragraphe doit être précédé de la commande `\item`. L'étiquette par défaut est un gros point, sinon c'est l'argument optionnel de `\item` qui est utilisé. Les environnements *itemize* peuvent être emboîtés. Par exemple :

```
\begin{itemize}
\item premier point
\item deuxi\`{e}me point
\item[3)] point avec \'{e}tiquette
\end{itemize}
```

produit :

- premier point
- deuxième point

3) point avec étiquette

4.7.2 enumerate

Produit le même résultat que *itemize*, mais les étiquettes par défaut sont des numéros automatiquement calculés.

4.7.3 quote et quotation

Utilisés pour les citations : on obtient une marge particulière.

4.7.4 tabbing

On dispose, dans cet environnement, de tabulations, à la façon d'une machine à écrire. La pose d'une tabulation se fait par la frappe de `\=`. Le passage à la ligne suivante par `\\`. Le saut à la tabulation suivante par `\>`. Si l'on désire se servir d'une ligne pour poser des tabulations, mais ne pas vouloir que la ligne apparaisse dans le texte final, il faut terminer la ligne par l'ordre `\kill`.
Par exemple, le source :

```
\begin{tabbing}
aaaaaaaaaaaaaaaaaaaaaa\=aaaaaaaaaaaaaaaaaaaa\=aaaaaaaaaaaaaaaaaaaaaa\= \kill
Perrine \> 8ans \> F \\
Nicolas \> 6ans \> M \\
Gaelle  \> 2ans \> F
\end{tabbing}
```

produit :

Perrine	8ans	F
Nicolas	6ans	M
Gaelle	2ans	F

4.7.5 tabular

Permet de créer des tableaux. Sa syntaxe est très proche de celle de l'environnement *array* (voir la description du mode mathématique). Il faut définir dès le départ le nombre de colonnes du tableau et leur justification, d'où l'argument $\{lcr\}$ dans l'exemple ci-dessous, qui signifie que la première colonne est justifiée à gauche, la seconde est centrée, tandis que la dernière est justifiée à droite. Lors de l'écriture du contenu du tableau, un & ordonne le passage à la colonne suivante, un \\ le passage à la ligne suivante, un `\multicolumn` permet de fusionner plusieurs colonnes. La commande `\hline` permet de tirer un trait entre deux lignes, tandis que des `|` insérés entre les caractères donnant la justification des colonnes permettent de séparer les colonnes par des traits verticaux.

Par exemple, ceci :

```
\begin{center}
\begin{tabular}{lc|r}
Nom & Note & Ville \\
Alice & 15 & Paris \\
Jules & 8 & Lyon \\
Nicolas & 7 & Marseille
\end{tabular}
\end{center}
```

donne :

Nom	Note	Ville
Alice	15	Paris
Jules	8	Lyon
Nicolas	7	Marseille

4.7.6 verbatim

L'environnement *verbatim* permet d'avoir en sortie une réplique exacte de ce qui se trouve dans le fichier source : les commandes sont ignorées, les blancs ne sont plus sautés ...

D'autres environnements sont décrits dans la suite de ce document comme par exemple l'environnement *picture* qui permet de créer des dessins et schémas simples, ou bien l'environnement *eqnarray* qui permet de créer des équations alignées.

4.8 Quelques commandes utiles particulières

- Pour empiler des mots `\shortstack`
- Pour tracer de jolis points de suspension `\ldots`
- Pour empêcher une indentation `\noindent`
- Pour créer une minipage `\begin`{*minipage*}... `\end`{*minipage*}
- Pour sortir sur deux colonnes `\twocolumn` puis revenir à une seule `\onecolumn`

4.9 Utilisation des boites

Une boite est une portion de texte que TEX traite comme une seule entité, une seule lettre. LATEX utilise très fréquemment cette structure implicitement, mais vous pouvez vous en servir directement pour certaines présentations.
La première commande, `\makebox`, avec comme argument le texte à mettre dans la boite, construit une boite de la taille du texte : cette boite ne peut être scindée. Vous pouvez toutefois donner en arguments optionnels la largeur de la boite que vous voulez construire ainsi que la justification du texte à l'intérieur de la boite. Exemple :

```
\makebox{Articles Sold\'{e}s\,:} \\
\begin{center}
\makebox[4cm][l]{Jupe}\makebox[3cm][l]{100Frs} \\
\makebox[4cm][l]{Pull}\makebox[3cm][l]{70Frs} \\
\makebox[4cm][l]{Sweat}\makebox[3cm][l]{85Frs} \\
\end{center}
```

donne

Articles Soldés

Jupe	100Frs
Pull	70Frs
Sweat	85Frs

La commande `\framebox` est équivalente, mais elle encadre en plus le contenu de la boite.
La commande `\parbox` permet de mettre en boite des paragraphes entiers. Il faut donner en premier argument la largeur de la boite, et le texte en second argument.
Enfin `\hbox` crée une boite horizontale, et `\vbox` une boite verticale.

4.10 Les références

4.10.1 Les références croisées

Pour pouvoir se référer à des formules ou à toute autre structure numérotée, il est utile de se servir des références relatives, car les références absolues peuvent changer si on modifie le texte. Pour cela on doit d'abord donner un nom à l'objet à référencer. C'est le rôle de la commande `\label`. Cette commande se place juste après les commandes de sectionnement ou après une

des commandes d'ouverture des environnements *equation*, *eqnarray*, *enumerate*, *figure*, *table* ou d'un environnement de type *newtheorem* (voir 6.1.3).
Pour appeler le numéro correspondant c'est la commande `\ref` qui doit être utilisée. Voici un exemple : le source

```
\subsection{Les r\'ef\'erences crois\'ees}
\label{refcrois}
\begin{equation}
{e}^{i\pi}+1=0
\label{eq:refcrois}
\end{equation}
```

```
La sous-section \ref{refcrois} contient l'\'equation \ref{eq:refcrois}
```

est traduit par :

$$e^{i\pi} + 1 = 0 \tag{4.1}$$

La sous-section 4.10.1 contient l'équation 4.1

On peut également ne donner que le numéro de la page avec la commande `\pageref`.

4.10.2 La bibliographie

C'est l'environnement *thebibliography* qui permet de taper toute la bibliographie. Il se comporte comme l'environnement *enumerate*. La commande `\bibitem` à l'intérieur de cet environnement se comporte à la fois comme la commande `\label` et la commande `\item`. D'autre part il existe une commande `\cite` analogue à `\ref` destinée à être immergée dans le texte. On dispose de quelques options qui se retrouvent dans l'exemple ci-dessous.

```
Les citations \cite{duval1,massias} peuvent \^etre sous la forme simple
pr\'ec\'edente ou bien sous une forme \cite[page 111]{duval2}
plus sophistiqu\'ee.
\begin{thebibliography}{99}
\bibitem{duval1} D. DUVAL. Cours de \LaTeX.
\bibitem{massias} J.-P. MASSIAS. Cours de \TeX.
\bibitem{duval2} D. DUVAL. Exercices de \LaTeX.
\end{thebibliography}
```

se traduit par :

Bibliographie

[1] D. DUVAL. Cours de LaTeX.

[2] J.-P. MASSIAS. Cours de TeX.

[3] D. DUVAL. Exercices de LaTeX.

Les citations [1, 2] peuvent être sous la forme simple précédente ou sous une forme [3, page 111] plus sophistiquée.

On peut également avoir les références sous l'autre forme classique [Duv 1] par exemple. Il est à noter qu'il faut traiter deux fois le source par la commande **latex** pour obtenir les citations dans le texte : la première fois, LaTeX construit un fichier contenant toutes les références, et, lors du second passage, les citations sont insérées.

4.11 Génération de tables ou d'index

Si vous spécifiez, dans le préambule, la commande `\tableofcontents`, vous obtiendrez une table des matières.

De même vous pourrez obtenir par `\listoffigures` ou `\listoftables` une table des figures ou une liste des tables. Il est aussi possible de modifier manuellement ces tables (qui sont dans les fichiers *nom-de-fichier.toc*, *nom-de-fichier.lof*, *nom-de-fichier.lot*). Vous devrez alors utiliser un `\nofiles` pour que ces versions modifiées ne soient pas écrasées lors d'un nouvel appel à LaTeX.

Il existe aussi un formateur de bibliographies, BibTeX, qui produit directement les `\bibitem` à partir de fichiers de références bibliographiques. Pour ceci, il faut indiquer dans le document l'endroit où se trouve le fichier de références, par la commande

```
\bibliography{mabiblio}
```

qui indique à LaTeX l'existence d'un fichier `mabiblio.bib`, BibTeX va extraire celles qui sont utilisées dans votre document par la commande `\cite`. La commande `\nocite` permet par ailleurs de faire apparaître dans la bibliographie une référence que l'on n'a pas citée dans le document.

Il faut aussi préciser le style de bibliographie souhaité, par la commande `\bibliographystyle`. On mettra par exemple

```
\bibliographystyle{plain}
```

Les styles existant en standard sont `plain`, `unsrt`, `alpha`, `abbrv`. Il peut en exister d'autres sur votre site, renseignez-vous auprès des administrateurs système qui doivent chercher les fichiers `.bst` dans le répertoire des librairies LaTeX.

Pour employer BibTeX, il faut d'abord exécuter une fois LaTeX sur le document, puis exécuter BibTeX, puis encore deux fois LaTeX.

En ce qui concerne le fichier suffixé `.bib`, qui contient la liste de vos références, il doit obéir à un format particulier. Les entrées ressemblent à:

```
@Article{ToTi91,
 author =    "Albert Toto and Bernard Titi",
 title =    "Exemple de titre avec une lettre accentu{\'e}e",
 journal =   "Bulletin de la Soci{\'e}t{\'e} Philharmonique",
 year =     1991,
 volume =    25,
 number =    3,
 pages =    "14--17",
 month =    jul
}
```

La première partie `@Article` indique qu'il s'agit d'un article de journal. Puis l'entrée est composée de *champs*, comme `title` dans l'exemple ci-dessus, dont certains sont obligatoires et d'autres optionnels. Les champs `author`, `title` et `year` sont presque toujours obligatoires et des champs comme `month` sont toujours optionnels. Il est néanmoins conseillé de mettre le plus d'information possible dans une référence. Voici pour chaque type d'entrée connu par BibTeX la liste de ses champs obligatoires (en caractères usuels) et optionnels (en italiques):

article author, title, journal, year, *volume, number, pages, month, note.*

book author (ou editor), title, publisher, year, *volume, series, address, edition, month, note.*

booklet title, *author, howpublished, address, month, year, note.*

inbook author (ou editor), title, chapter, pages, publisher, year, *volume, series, address, edition, month, note.*

incollection author, title, booktitle, publisher, year, *editor, chapter, pages, address, month, note.*

inproceedings author, title, booktitle, year, *editor, pages, organization, publisher, address, month, note.*

manual title, *author, organization, address, edition, month, year, note.*

mastersthesis author, title, school, year, *address, month, note.*

misc *author, title, howpublished, month, year, note.*

phdthesis comme mastersthesis.

proceedings title, year, *editor, publisher, organization, address, month, note.*

techreport author, title, institution, year, *type, number, address, month, note.*

unpublished author, title, note, *month, year.*

Chapitre 5

Le mode mathématique

5.1 Sa déclaration

Ce formateur de textes a l'avantage de disposer de toute une batterie de commandes permettant un bon rendu de textes mathématiques. Pour pouvoir utiliser ces commandes, il faut sortir du mode standard pour entrer dans le mode dit mathématique : le changement de mode peut se déclarer de trois manières différentes selon votre but :

- Par un `\begin`{*math*}...`\end`{*math*}, (simplifié par \$...\$ ou par `\(` ... `\)`), pour insérer une formule mathématique dans une ligne de texte.
- Par un `\begin`{*displaymath*}...`\end`{*displaymath*}, (simplifié `\[...\]`) pour des formules isolées et centrées sur une ligne.
- Par un `\begin`{*equation*}...`\end`{*equation*}, qui produit le même résultat mais numérote automatiquement les formules.

Attention LaTeX gère lui-même les espaces. Il les répartit au mieux de ce qu'il juge être correct. Si vous désirez absolument des blancs à des endroits précis, il faudra les forcer.

5.2 Son jeu de caractères

5.2.1 Les lettres grecques

Elles sont obtenues en ajoutant un \ au nom de la lettre. Pour obtenir les majuscules, il suffit de taper la première lettre en majuscule.

Exemple :	α	`\alpha`
	Γ	`\Gamma`

5.2.2 Les lettres cursives

Pour les obtenir il faut appeler une fonte spéciale par la déclaration `\cal`
Exemple : Soit une fonction $\mathcal{F}$ avec $\mathcal{F}(x) > 0$.
se code en
Soit une fonction `$\cal F$ avec ${\cal F} (x) > 0$`.

5.2.3 Autres symboles

On trouve dans LaTeX tout un jeu de symboles mathématiques dans lequel on peut distinguer :

Symboles de type sommation et produit

Ces symboles acceptent un indice et un exposant, qui seront respectivement placés au-dessous et au-dessus du symbole. Par exemple

```
\sum_{i=0}^{n-1} a_i
```

produit

$$\sum_{i=0}^{n-1} a_i$$

dans l'environnement *displaymath* et $\sum_{i=0}^{n-1} a_i$ en environnement *math*.

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

Les opérateurs binaires

$\pm$ `\pm`	$\cap$ `\cap`	$\diamond$ `\diamond`	$\oplus$ `\oplus`
$\mp$ `\mp`	$\cup$ `\cup`	$\bigtriangleup$ `\bigtriableup`	$\ominus$ `\ominus`
$\times$ `\times`	$\uplus$ `\uplus`	$\bigtriangledown$ `\bigtriangledown`	$\otimes$ `\otimes`

$\div$ `\div`	$\sqcap$ `\sqcap`	$\triangleleft$ `\triangleleft`	$\oslash$ `\oslash`
$\ast$ `\ast`	$\sqcup$ `\sqcup`	$\triangleright$ `\triangleright`	$\odot$ `\odot`
$\star$ `\star`	$\vee$ `\vee`	$\lhd$ `\lhd`	$\bigcirc$ `\bigvirc`
$\circ$ `\circ`	$\wedge$ `\wedge`	$\rhd$ `\rhd`	$\dagger$ `\dagger`
$\bullet$ `\bullet`	$\setminus$ `\setminus`	$\unlhd$ `\unlhd`	$\ddagger$ `\ddagger`
$\cdot$ `\cdot`	$\wr$ `\wr`	$\unrhd$ `\unrhd`	$\amalg$ `\amalg`

Les opérateurs de relation

$\leq$ `\leq`	$\geq$ `\geq`	$\equiv$ `\equiv`	$\models$ `\models`
$\prec$ `\prec`	$\succ$ `\succ`	$\sim$ `\sim`	$\perp$ `\perp`
$\preceq$ `\preceq`	$\succeq$ `\succeq`	$\simeq$ `\simeq`	$\mid$ `\mid`
ll `ll`	$\gg$ `\gg`	$\asymp$ `\asymp`	$\parallel$ `\parallel`
$\subset$ `\subset`	$\supset$ `\supset`	$\approx$ `\approx`	$\bowtie$ `\bowtie`
$\subseteq$ `\subseteq`	$\supseteq$ `\supseteq`	$\cong$ `\cong`	$\Join$ `\Join`
$\sqsubset$ `\sqsubset`	$\sqsupset$ `\sqsupset`	$\neq$ `\neq`	$\smile$ `\smile`
$\sqsubseteq$ `\sqsubseteq`	$\sqsupseteq$ `\sqsupseteq`	$\doteq$ `\doteq`	$\frown$ `\frown`
$\in$ `\in`	$\ni$ `\ni`	$\propto$ `\propto`	
$\vdash$ `\vdash`	$\dashv$ `\dashv`		

Les opérateurs fléchés

$\leftarrow$ `\leftarrow`	$\longleftarrow$ `\longleftarrow`	$\uparrow$ `\uparrow`
$\Leftarrow$ `\Leftarrow`	$\Longleftarrow$ `\Longleftarrow`	$\Uparrow$ `\Uparrow`
$\rightarrow$ `\rightarrow`	$\longrightarrow$ `\longrightarrow`	$\downarrow$ `\downarrow`
$\Rightarrow$ `\Rightarrow`	$\Longrightarrow$ `\Longrightarrow`	$\Downarrow$ `\Downarrow`
$\leftrightarrow$ `\leftrightarrow`	$\longleftrightarrow$ `\longleftrightarrow`	$\updownarrow$ `\updownarrow`
$\Leftrightarrow$ `\Leftrightarrow`	$\Longleftrightarrow$ `\Longleftrightarrow`	$\Updownarrow$ `\Updownarrow`
$\mapsto$ `\mapsto`	$\longmapsto$ `\longmapsto`	$\nearrow$ `\nearrow`
$\hookleftarrow$ `\hookleftarrow`	$\hookrightarrow$ `\hookrightarrow`	$\searrow$ `\searrow`
$\leftharpoonup$ `\leftharpoonup`	$\rightharpoonup$ `\rightharpoonup`	$\swarrow$ `\swarrow`
$\leftharpoondown$ `\leftharpoondown`	$\rightharpoondown$ `\rightharpoondown`	$\nwarrow$ `\nwarrow`
$\rightleftharpoons$ `\rightleftharpoons`	$\leadsto$ `\leadsto`	

Divers symboles

$\aleph$ `\aleph`	$\prime$ `\prime`	$\forall$ `\forall`	∞ `\infty`
$\hbar$ `\hbar`	$\emptyset$ `\emptyset`	$\exists$ `\exists`	$\Box$ `\Box`
$\imath$ `\imath`	∇ `\nabla`	$\neg$ `\neg`	$\Diamond$ `\Diamond`
$\jmath$ `\jmath`	$\surd$ `\surd`	$\flat$ `\flat`	$\triangle$ `\triangle`
ℓ `\ell`	$\top$ `\top`	$\natural$ `\natural`	$\clubsuit$ `\clubsuit`

$\wp$ \wp	$\perp$ \bot	$\sharp$ \sharp	$\diamondsuit$ \diamondsuit
$\Re$ \Re	$\|$ \\|	$\backslash$ \backslash	$\heartsuit$ \hearsuit
$\Im$ \Im	$\angle$ \angle	∂ \partial	$\spadesuit$ \spadesuit
$\mho$ \mho			

Les fonctions mathématiques

En style mathématique, log représente le produit l, o, g et pour indiquer la fonction elle-même il faut taper la commande `\log`. On trouve les fonctions mathématiques usuelles :

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

Exemple :

```
\( \log xy = \log x + \log y \)
```

produit :
$\log xy = \log x + \log y$

5.3 Construction des formules mathématiques

5.3.1 Les indices et les exposants

Les indices sont indiqués par le symbole _ et ^ indique un exposant.
Exemples :

x^2 `$x^2$`	x^{2y} `$x^{2y}$`	x^{y^2} `$x^{2^{y}}$`
x^y_1 `$x^{y}_{1}$`	x^{y_1} `$x^{y_{1}}$`	$x^{y^{x_1}}$ `$x^{y^{x_{1}}}$`

5.3.2 Les fractions

Elles sont obtenues en ligne par `/`, et par le symbole `\frac` dans les formules à plusieurs niveaux.

Exemple :
`\( (x+y+z)/z \)` donne : $(x+y+z)/y$.

```
\[ a=frac{b+c/2}{c^{2}+1} \]
```

donne :

$$a = \frac{b+c/2}{c^2+1}$$

5.3.3 Les tableaux

Ils sont produits grâce a l'environnemet *array*. Il faut donner en argument le nombre de colonnes et leur justification (**c** pour centrage, **l** pour justification à gauche, **r** pour justification à droite).
Chaque ligne du tableau doit se terminer par **** et les colonnes sont séparées par &.
Exemples :

$$\begin{array}{crcl}
a+b+c & u+v+w & x & 468 \\
a+b & uv & x-y-z & 68 \\
b & 3u+5v & y & 54
\end{array}$$

```
\begin{center}
\( \begin{array}{crcl}
a+b+c & u+v+w & x & 468 \\
a+b   & uv & x-y-z & 68 \\
b & 3u+5v & y & 54
\end{array} \)
\end{center}
```

Les systèmes d'équations et les formules multi-lignes

Les environnements *displaymath* et *equation* n'autorisent la construction que d'une seule formule. Aussi pour écrire des systèmes d'équations ou d'inéquations, on utilisera l'environnement *eqnarray*. C'est un tableau à trois colonnes dont les lignes seront séparées par **** et les colonnes par des &. Les lignes seront numérotées.
On peut utiliser l'environnement *eqnarray** si l'on ne désire aucune numérotation.

Exemple :

$$\begin{eqnarray} ax+by+c &=& 2 \\ ax-1 &>& 3 \end{eqnarray}$$

(5.1)
(5.2)

est produit par

```
\begin{eqnarray}
ax+by+c &= & 2 \\
ax-1    & > & 3
\end{eqnarray}
```

5.3.4 Les délimiteurs

Ils jouent le même rôle que les parenthèses dans les expressions mathématiques. LaTeX en reconnait un certain nombre.

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

Pour pouvoir utiliser un délimiteur dont la taille soit correctement calculée, il faut le faire précéder d'un `\left` ou d'un `\right`. Les délimiteurs devront toujours aller par deux. Dans le cas où le second doit être invisible, il faudra marquer sa place par un point.
Exemples :

$$\left(\begin{array}{c} \left| \begin{array}{cc} x^{11} & x^{12} \\ x^{21} & x^{22} \end{array} \right| \\ y \\ z \end{array}\right)$$

est produit par :

```
\[\left(\begin{array}{c}
          \left| \begin{array}{cc}
          x^{11} & x^{12} \\
```

```
            x^{21} & x^{22}
                   \end{array}
            \right| \\
            y \\ z
        \end{array}\right) \]
```

tandis que :

$$a^n = \left\{ \begin{array}{ll} 1 & \mbox{si } n=0 \\ aa^{n-1} & \mbox{si } n>0 \end{array} \right.$$

vient de :

```
\[{a}^{n}=\left{\begin{array}{ll}
               1 & \mbox{si $n=0$} \\
               a{a}^{n-1} & \mbox{si $n>0$
             \end{array}
         \right. \]
```

5.3.5 Les accents et les symboles empilés

Les accents

Il est peu courant de rencontrer des accents en mode mathématique. Toutefois la notation habituelle n'étant pas autorisée, il faut utiliser les commandes suivantes : (l'exemple est donné avec la lettre a comme référence mais est valable quelque soit celle-ci)

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

Les traits horizontaux

Pour souligner une partie de formule il faut utiliser la commande `\underline`. De même, si l'on désire tracer un trait horizontal au-dessus de certains symboles ou groupe de symboles, on utilise `\overline`.
En outre, pour tracer des accolades horizontales au-dessus ou au-dessous d'une formule `\overbrace` et `\underbrace` seront employées.
Exemple : $\underbrace{x+y+z+\overbrace{u+v+w}}$ provient de : `\underbrace{x+y+z+ \overbrace{u+v+w}}`

Les symboles empilés

Si l'on désire placer un symbole au-dessus d'un autre, on utilisera `\stackrel` (le symbole supérieur est alors de taille réduite).
Exemple : $X \stackrel{x'}{\rightarrow} Y$ `\( X \stackrel{x'}{\rightarrow} Y \)`

5.3.6 Les espaces

En général, TEX se débrouille tout seul pour placer au mieux les blancs dans les formules. Vous pouvez toutefois si le résultat ne vous convient pas, en ajouter vous-mêmes. Pour cela vous disposez de plusieurs dimensions d'espacement :

- `\,` éloignera légèrement deux caractères
- `\!` rapprochera légèrement deux caractères
- `\:` espacera deux caractères
- `\;` éloignera deux caractères

5.3.7 Changement de style en mode mathématique

Le style utilisé par défaut est l'italique, légèrement différent de l'italique du texte.
Vous pouvez toutefois utiliser `\cal` pour les cursives. Si vous voulez des caractères gras, il suffit de spécifier `\boldmath`.

Enfin, si le mode mathématique de LATEX ne vous donne pas satisfaction pour la rédaction de vos formules, vous aurez trois choix possibles :

- soit vous vous plongez dans TEX.
- soit vous dessinez vos formules à l'aide du mode graphique.
- soit vous demandez à quelqu'un qui sait.

On remarquera qu'il existe de nombreuse fontes utiles (gothiques ou autres) dont la disponibilité dépend des sites : on n'en parlera pas ici.

Chapitre 6

Comment définir vos propres présentations

Si vous fabriquez vos propres présentations, il existe un certain nombre d'outils que nous allons essayer de passer en revue.

6.1 Définition de nouvelles commandes et de nouveaux environnements

6.1.1 Nouvelles commandes

Si vous désirez utiliser les mêmes structures plusieurs fois dans votre texte, vous pouvez créer vos propres commandes. La définition d'une commande personnelle se place juste après le choix du style de document et avant la commande `\begin`{*document*}.

Pour définir une nouvelle commande, il faut utiliser la commande `\newcommand` qui a deux arguments : le nom de la commande et son contenu.

Exemple :

```
\newcommand{\labo}{{\large \'{E}cole Polytechnique}}
```

a pour effet à chaque appel de `\labo` de faire imprimer en gros caractères le nom complet du labo.

Exemple :

```
\labo
```

produit

École Polytechnique.

Il est souvent utile de se définir de nouvelles commandes pour obtenir des abréviations de commandes ou d'environnements fréquemment utilisés.

Exemple :

```
\newcommand{\bn}{\begin{enumerate}}
\newcommand{\en}{\end{enumerate}}
```

Toutefois, dans certains cas, vous pouvez avoir besoin de variables dans vos définitions. Il est possible d'utiliser jusqu'à neuf variables par commande. Leur nombre sera donné en argument optionnel : le symbole $\#n$ (où n est un entier) désigne le n-ième argument de la commande dans la définition de la commande. Lors de l'appel de la commande, les paramètres seront les arguments de la commande. Par exemple, la ligne suivante définit une commande *tc* à deux arguments :

```
\newcommand{\tc}[2]{{\bf {#1}} {\it {#2}}}
```

qui produit : **latex** *nom-de-fichier.tex*
à partir de l'appel :

```
\tc{latex}{nom-de-fichier.tex}
```

Si vous désirez redéfinir une commande standard ou personnelle, il vous faudra utiliser la commande
`\renewcommand` au lieu de `\newcommand`.

6.1.2 Définition de nouveaux environnements

Le même mécanisme de définition existe en ce qui concerne les environnements. Il suffit d'appeler la commande :

```
\newenvironment{nom-de-l'environnement}{begin text}{end text}
```

En général on s'appuie sur un environnement existant auquel on ajoute certaines spécificités. On peut de même que dans la définition de commandes utiliser des variables.

Exemple :

```
\newenvironment{itlist}{\begin{enumerate} \it}{\end{enumerate}}
```

produira une liste numérotée dont les éléments seront en italique.

6.1.3 Les théorèmes

Souvent des textes mathématiques font mention de théorèmes ou de structures tels que lemmes, axiomes ... Des textes non mathématiques peuvent avoir besoin de ce genre de structures pour définir des lois, principes, règlements ... Vous pouvez, grâce à la commande `\newtheorem`, présenter ces structures. Cette commande possède deux arguments, le premier est le nom de l'environnement, le second est le texte de l'étiquette.

Exemple :

La commande :

```
\newtheorem{reg}{Article}
```

définit la structure *reg*.

Le source :

```
\begin{reg}
Tous les locaux doivent \^{e}tre soigneusement ferm\'{e}s
\end{reg}
```

produit :

Article 1 *Tous les locaux doivent être soigneusement fermés.*

La numérotation est automatique.

6.2 Les dessins

L'environnement *picture* permet de dessiner des figures composées de texte, de segments de droites, de cercles. Une figure se construit toujours de la même manière :

```
\begin{picture}(largeur,hauteur)
....commandes relatives au dessin......
\end{picture}
```

Tout est calculé par rapport à la valeur de `\unitlength` (par défaut elle vaut 1pt soit 1/72 inch). Pour changer cette valeur on utilisera `\setlength`.

6.2.1 Les ordres relatifs au dessin

Tout dessin est construit à l'aide de la commande de positionnement `\put`(x,y){*objet*}
L'*objet* peut être soit

- du texte
- une boite
- une droite
- une flèche
- une pile
- un cercle
- un ovale ou un rectangle arrondi

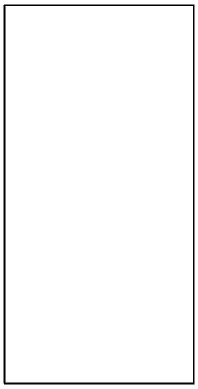
Remarques :

- Dans le cas des boites, les commandes utilisées sont les mêmes que celles vues auparavant, mais ont une syntaxe légèrement différente. Il faut toujours spécifier la largeur et la hauteur de la boite.
- Le tracé d'une droite se fait par `\line`(x,y){*longueur*}. Les paramètres x,y donnent la direction dans laquelle se trace la droite, puis l'on doit préciser la longueur.
- Une droite fléchée se trace par la commande `\vector`
- Pour empiler du texte vous utiliserez `\shortstack`

- Un cercle se trace par la commande `\circle`
- Un disque noir se dessine par la commande `\circle*` En argument il faut donner le rayon.
- Un ovale est un rectangle aux angles arrondis dont on donne la largeur et la hauteur, le point de référence étant le centre de l'ovale. Il est possible en donnant des arguments complémentaires de ne tracer qu'une partie de l'ovale.

Enfin pour ne pas répéter des séquences entières permettant de tracer une partie de figure, on peut regrouper un certain nombre de commandes répétitives sous une commande `\multiput`

Exemple :



est produit par

```
\setlenght{\unitlength}{0.5cm}
\begin{picture}(10,10)
\put(0,0){\line(1,0){5}}
\put(5,0){\line(0,1){10}}
\put(5,10){\line(-1,0){5}}
\put(0,10){\line(0,-1){10}}
\end{picture}
```

Remarquons qu'en-dessous d'une taille limite (10pts ou 1/7 inch), LaTeX ne trace pas de segments de droite.

6.2.2 Les autres outils pour dessiner

Les utilitaires PostScript

Il existe un utilitaire nommé Psfig, du domaine public, qui permet d'inclure très simplement des fichiers PostScript, tels que ceux créés par des logiciels tels que MacDraw, MacPaint, Adobe Illustrator (sur Macintosh) ou encore Xfig, Idraw (sous X-Window).

Pour include un fichier PostScript, il suffit de taper dans le début de son document la commande `\input`{*psfig*}, puis à l'endroit voulu, on tape `\psfig`{*figure=un-fichier-Postcript*}. De plus, il est possible de donner à la figure une taille arbitraire. Une documentation est livrée avec ce package.

Les utilitaires TeX

Pour des figures plus compliquées, l'utilitaire PiCTeX, lui aussi du domaine public, permet de réaliser des figures géométriques assez sophistiquées. De même que précédemment, il faut inclure le fichier de macros par `\input`{*prepictex*}, `\input`{*pictex*}, et `\input`{*postpictex*} au début du document, puis taper :

```
\beginpicture
        des commandes PiCTeX
\endpicture
```

Pour inclure des dessins réalisés par des outils X-Window tels que Xfig ou GnuPlot, on peut également utiliser la boite à outils TransFig. Elle permet la transformation de dessins au format Fig aux formats LaTeX PostScript, PiCTeX, PIC. L'utilisation sous LaTeX est alors du type :

```
\documentstyle{article}
\begin{document}

\begin{figure}
\begin{center}
\input(fichier-de-figure}
\end{center}
\end(figure}
```

6.3 Les styles

Les commandes et la déclaration d'un préambule qui sont fréquemment utilisées peuvent être regroupées dans un fichier portant un nom de la forme : *nom-de-style.sty*. Il suffit alors d'ajouter *nom-de-style* à la liste des options de la commande `\documentsyle` de votre préambule pour que le fichier *nom-de-style.sty* soit automatiquement chargé.

Et maintenant, à vous de jouer ! Bon courage !