Aide mémoire pour LaTeX

Par C. Willems et F. Geraerds.
Etudiants à l'Université de Liège, Belgique.

Dernière révision: 4 juillet 1996.

Ce document comporte ce qu'il est absolument indispensable de connaître pour débuter avec LaTeX. Ce document ne constitue qu'une introduction au logiciel et ne remplace en aucun cas le guide de référence proposé par Leslie Lamport ("*A document preparation system, User's guide & reference manual*").

1 Fonctionnement

LaTeX est un traitement de texte prenant un fichier d'entrée en format texte. Pour éditer un fichier compréhensible, il suffit d'utiliser un éditeur de textes.

Les seuls caractères autorisés dans les fichiers d'entrée sont les suivants: les lettres minuscules et majuscules, les chiffres et les 16 caractères de ponctuation

. ; , ? : ! ' ` ( ) [ ] - / * @

Les 10 caractères suivants sont utilisés pour les commandes LaTeX.

$ & \ % _ ^ ~ { }

Les 5 caractères

+ = < > |

sont utilisés uniquement en mode mathématique bien que les deux premiers puissent être utilisés en mode normal. Le caractère `"` n'est jamais utilisé.

Un fichier d'entrée pour LaTeX porte généralement l'extension `.tex`. Ceci n'est pas une obligation mais il est vivement conseillé de respecter cette règle qui est très pratique pour distinguer les fichiers manipulés.

Lorsque le fichier d'entrée est complet (càd. qu'il respecte la syntaxe de LaTeX), il peut être compilé pour obtenir un second fichier qui porte le même nom que le premier, au fait près que l'extension est `.dvi` (*device-independent*). D'autres fichiers sont également créés et portent les extensions `.log, .aux, .toc, .lof, .lot,`... Seul le fichier `.log` est utile pour l'utilisateur. Celui-ci contient le rapport de la compilation du fichier d'entrée. On y trouve notamment les erreurs et leurs positions (lignes dans le fichier d'entrée). Les autres fichiers ne peuvent être modifiés sous peine de provoquer des erreurs lors de la compilation suivante. Tous peuvent être supprimés du disque une fois le texte terminé.

Le fichier `.dvi` peut être visualisé et/ou imprimé à l'aide de logiciels adéquats (`dviscr, dvips, dvihplj,`...) fournis indépendamment de LaTeX.

2 Nouveau document

La forme générale d'un document LaTeX est

\documentstyle[*options*]{*style*}
preambule
\begin{document}
texte
\end{document}

où

style représente le style du document. Les principaux styles utilisés par LaTeX sont `book`, `report`, `article` et `letter`.

options désigne une liste d'une ou plusieurs options séparées par des virgules. On peut spécifier la taille des caractères (`11pt`, `12pt` ou par défaut 10 points), des fichiers de styles (dont on parle plus bas),...

preambule contient les définitions éventuelles de commandes, d'environnements,... (voir plus bas)

texte est le texte (en format LaTeX).

2.1 La syntaxe LaTeX

Nous avons donné plus haut les caractères autorisés dans le texte fourni au compilateur. De nombreux autres caractères sont cependant disponibles mais s'obtiennent par des *commandes* LaTeX. Il est très important de respecter l'écriture (majuscules ou minuscules) des commandes car LaTeX en tient compte.

Par exemple, les caractères accentués sont d'utilité courante en français et sont obtenus à l'aide des commandes données dans le tableau 1. De même, les lettres grecques, les symboles mathématiques,... sont obtenus par les commandes données à la fin du document. Pour les accents, il faut juste préciser que pour mettre un accent sur le `i`, il ne faut pas utiliser le i normal qui possède un point mais le `\i` (ı) qui ne possède pas de point. Ainsi î et obtenu par `\^{\i}`.

Des commentaires peuvent être insérés dans le fichier d'entrée. Ceux-ci n'apparaissent pas dans le texte finale. Tout texte compris entre le caractère % (non directement précédé par un \) et la fin de ligne dans un fichier d'entrée est considéré comme un commentaire.

Les caractères réservés #, $, \, &, %, { et } s'obtiennent grâce aux commandes respectives `\#`, `\$`, `$\backslash$`, `\&`, `\%`, `\{` et `\}`

2.2 Commande

Lorsqu'un texte apparaît à plusieurs reprise dans un document, il est intéressant d'abréger ce texte à l'aide d'un mot clé. L'instruction

`\newcommand`{*nom*}{*texte*}

associe le texte au nom de commande. Chaque fois que ce nom apparaît, c'est le texte associé qui sera imprimé. Le nom doit commencer par le caractère réservé \ et ne peut pas avoir été défini préalablement. LaTeX ignore l'espace qui suit une commande. Ainsi pour laisser un espace après une commande, il faut utiliser `\␣`.

Il est possible de fournir des paramètres à une commande pour générer des blocs de textes dont seules des parties varient. L'instruction

`\newcommand`{*nom*}`[`*nargs*`]`{*texte*}

définit une commande qui possède *nargs* arguments (ce nombre est fixé à la définition). Dans le texte, on y a accès par le caractère `#` immédiatement suivi de la position de l'argument (ex. `#1`).

Exemple 1 *L'ensemble des réels* $\mathbb{R}$ *s'obtient par les instructions* `{\rm I\!R}`. *On définit la commande de nom* `\R` *pour cet ensemble par*

`\newcommand{\R}{{\rm I\!R}}`

2.3 Environnement

Un environnement est une mise en forme particulière d'un texte comme par exemple une énumération, une description,... Pour débuter un environnement *nom*, on utilise l'instruction

`\begin`{*nom*}

et pour terminer, l'instruction

`\end`{*nom*}

De nombreux environnements sont prédéfinis par LaTeX. Par exemple, `array`, `itemize`, `enumerate`, `description`, `tabular`, `eqnarray`, `equation`,... Plusieurs seront évoqués dans la suite de ce document (cf. 3.9).

L'instruction

`\newenvironnement`{*nom*}{*beg_texte*}{*end_texte*}

définit un nouvel environnement. Celui-ci débutera par *beg_texte* et se terminera *end_texte*. En d'autres termes, l'instruction `\begin`{*nom*} provoquera l'affichage de *beg_texte* et l'instruction `\end`{*nom*} l'affichage de *end_texte*. Comme pour les commandes, il est possible de définir des environnements qui reçoivent des arguments.

2.4 Théorème

Les textes mathématiques contiennent souvent des théorèmes et des structures similaires telles que axiomes, définitions, lemmes, propositions,... Avoir un environnement particulier pour chaque structure est impensable. C'est pourquoi LaTeX propose la commande `\newtheorem` pour de telles structures. Cette commande possède deux arguments. Le premier est le nom de l'environnement (ici `The`) et le second est le texte pour l'identifier (ici `Théorème`).

```
\newtheorem{The}{Théorème}
```

Les différentes structures sont numérotées individuellement. On peut soumettre la numérotation à celle des parties d'un document comme les chapitres. Pour cela on utilise un troisième argument entre crochet.

```
\newtheorem{The}{Théorème}[chapter]
```

Si on désire que plusieurs structures soient numérotées ensembles, on définit par exemple

```
\newtheorem{The}{Théorème}
\newtheorem{Pro}[The]{Proposition}
```

et la proposition qui suit le théorème 5 sera numérotée par 6.

2.5 Fichier de style

Les macros (commandes, environnements et théorèmes) sont généralement utilisées dans de nombreux documents. Les placer dans le préambule de chaque document n'est pas pratique. On peut les regrouper dans un fichier annexe appelé *fichier de style* et dont l'extension est `.sty`. Ces fichiers ne répondent pas à la structure définie plus haut mais contiennent les instructions les unes à la suite des autres.

Il est par exemple très pratique de créer un document qui contient les intitulés français des différentes parties (si on ne possède pas un style français). De même, créer un fichier qui contiendra les déclarations de taille des pages et un troisième qui contiendra les macros personnelles.

3 Mise en page

Les mises en pages sont automatiques et demandées à l'aide de commandes dans le fichier source.

LaTeX ignore les espaces multiples ainsi que les fins de ligne dans le fichier d'entrée. Par exemple, entre deux mots, placer un espace ou dix, cela revient au même lors de l'affichage. (Il est dès lors possible d'indenter le fichier source, de faire des lignes longues ou courtes,...).

3.1 Paragraphe

Un praragraphe est obtenu en insérant une ligne vide dans le fichier d'entrée. (Une ligne vide est une ligne qui ne contient aucun caractère visible mais qui peut contenir des espaces ou des tabulations.)

3.2 Coupures des lignes et pages

Il arrive que LaTeX termine une ligne ou une page en des endroits inappropriés comme au milieu d'un mot, d'un paragraphe qui ne peut l'être pour l'une ou l'autre raison. Il propose plusieurs commandes pour permettre un certain contrôle à l'utilisateur.

`\linebreak[num]` et `\nolinebreak[num]`
: La commande `\linebreak[num]` encourage et `\nolinebreak[num]` décourage une coupure de ligne suivant `num` qui est un nombre de 0 à 4 (défaut 4). Une valeur de 4 oblige la coupure ou l'interdit. Un avertissement (`underfull hbox`) se produit si trop d'espace entre les mots résulte de cette commande.

`\\[len]` et `\newline`
: Les deux commandes provoquent une coupure de ligne sans justification de la ligne sur laquelle cette coupure se produit. Le texte est aligné à gauche. Le paramètre optionel `len` permet d'ajouter

un espace vertical supplémentaire entre les lignes. Ces commandes ne peuvent se trouver en fin de paragraphe sinon elles provoquent un avertissement.

`\-`
: Utilisée dans un mot, cette commande autorise LaTeX à couper le mot à cet endroit. Le mot ne peut être coupe en un autre endroit. Plusieurs commandes peuvent se trouver dans un même mot. La coupure dans le texte final est marquée par un trait d'union.

`\hyphenation{mots}`
: Déclare une liste de mots pouvant être coupés. Les coupures autorisées sont indiquées par des tirets (-). Cette déclaration est globale (valable dans tout le document à partir du point où elle se situe).

`\fussy` et `\sloppy`
: Ces deux commandes définissent la manière dont LaTeX réagit lorsqu'il se trouve face à un `overfull hbox`. La valeur par défaut est `\fussy`. Dans ce cas, si le texte est trop long pour tenir sur la ligne et qu'une coupure provoquerait un avertissement, le texte est laissé sur la ligne et dépasse dans la marge. Dans l'autre cas, le texte est mis à la ligne mais produit un avertissement. La manière dont un paragraphe est compilé dépend de la valeur en fin de celui-ci.

`\begin{sloppypar}`*pars*`\end{sloppypar}`
: Compile le(s) paragraphe(s) *pars* avec l'option `\sloppy`.

`\pagebreak[num]` et `\nopagebreak[num]`
: La commande `\pagebreak[num]` facilite et la commande `\nopagebreak[num]` décourage la coupure d'une colonne suivant le paramètre `num` (nombre de 0 à 4, défaut 4). Lorsqu'elles sont utilisées dans un paragraphe, elles s'appliquent à la fin de la ligne dans laquelle elles apparaissent.

`\samepage`
: La commande `\samepage` prévient d'une coupure de page dans certains cas (voir guide de référence p. 90).

`\clearpage`
: La commande `\clearpage` oblige LaTeX à afficher tous les tableaux et figures qui ne le sont pas encore et à débuter une nouvelle page.

`\cleardoublepage`
: La commande `\cleardoublepage` agit comme `\clearpage` et en plus, dans le cas d'un texte sur deux faces, force le passage à une page de droite.

`\newpage`
: La commande `\newpage` force le passage à une nouvelle colonne mais ne force pas l'affichage des tableaux et figures.

3.3 Les espaces

LaTeX ignore les espaces multiples dans le fichier d'entrée (sauf dans l'environnement `verbatim`). Les commandes suivantes obligent LaTeX à passer plus d'espace. Pour cela, il faut définir une unité de mesure. Nous ne considérerons que deux cas: le millimètre (`mm`) et le point (`pt`). Une longueur est un nombre suivi directement et sans espace d'une unité de mesure. On utilise aussi `1ex` pour *une fois la longueur de la lettre x dans la police et taille courante.*

`\hspace{esp}`, `\hspace*{esp}`
: Produit un espace horizontal de `esp` (une longueur positive ou négative). Dans le premier cas, cet espace disparaît s'il survient à une coupure de ligne, dans le second pas.

`\vspace{exp}` et `\vspace*{esp}`
: Produit un espace vertical de `esp` (longueur positive ou négative). Si cette commande apparaît dans un paragraphe, l'espace est passé en dessous de la ligne dans laquelle elle apparaît. L'espace disparaît dans le premier cas s'il se produit à une coupure de colonne.

`\smallskip`, `\medskip` et `\bigskip`
: Ces commandes sont des raccourcis pour des `\hspace{...}` avec des longueurs prédéfinies.

On consultera le guide pour plus d'informations sur les longueurs, notamment pour associer un nom à une longueur.

La longueur fournie aux commandes ci-dessus est fixe. Il est parfois utile d'avoir une longueur extensible en ce sens qu'elle prend le plus d'espace possible. La commande `\fill` agit comme cela. La commande `\hfill` est un raccourci pour `\hspace{\fill}`. Son effet est simple. Ce qui se trouve après est mis le plus à droite possible, donc contre la marge de gauche dans un paragraphe habituel. Si deux de ces commandes se trouvent sur une même ligne, l'élément du milieu se trouve au centre de l'espace qu'il y a entre les deux éléments extérieurs. La commande `\vfill` est un raccourci pour `\vspace{\fill}` et agit de manière similaire.

La commande `\dotfill` (`\hrulefill`) agit comme `\hfill`, au fait près qu'elle remplit l'espace par des pointillés (une ligne pleine).

3.4 Les espaces entre paragraphes et entre lignes

Par défaut, LaTeX n'insère pas d'espace supplémentaire entre deux paragraphes ni entre deux lignes. L'espace laissé est minimum. Cependant il est possible de le changer. La longueur `\parskip` correspond à l'espace entre deux paragraphes. Elle ne peut être modifiée que dans le *preambule*. La longueur `\baselineskip` correspond à l'espace entre les lignes d'un même paragraphe et peut être modifiée n'importe où.

Pour modifier une longueur, on tape simplement le nom de la longueur suivi d'un = et de la nouvelle longueur. Par exemple `\parskip = 5pt`

3.5 Chapitres, sections, ... et annexes

Un texte est généralement subdivisé en de nombreuses parties (numérotées le plus souvent). LaTeX propose pour cela 7 niveaux qui sont

```
\part          \subsection          \paragraph
\chapter       \subsubsection       \subparagraph
\section
```

et s'utilisent en mettant l'intitulé entre accolades après la commande. La première commande (`\part`) provoque la division du document en parties sans altérer les numérotations. La commande `\chapter` provoque un `\cleardoublepage`, l'affichage du mot `Chapitre` (en anglais si le style l'est) suivit du numéro et de l'intitulé. Les trois commandes suivantes provoquent des numérotations successives et en cascade de même qu'elles utilisent des styles d'écriture gras et plus grands que le texte normal. Les deux dernières ne sont pas numérotées. L'intitulé est en gras et indenté pour la deuxième. La commande `\appendix` redéfinit le nom associé à la commande `\chapter`. Ce nom correspond à une annexe. Donc toute commande `\chapter` placée après la commande `\appendix` provoque l'affichage du mot *Annexe* (*Appendix* en anglais) au lieu de *Chapitre*.

3.6 Table des matières

Créer une table des matières avec LaTeX est très simple. Il suffit d'insérer dans le texte la commande `\tableofcontents`. La table contient toutes les parties numérotées (parties, chapitres, annexes, sections et sous-sections). Elle commence une nouvelle page dans le style `book` et `report`. Elle débute par le titre *Table des matières*.

3.7 Notes de bas de pages

Pour insérer une note de bas de page avec LaTeX il suffit d'insérer dans le texte, à coté du mot clé, la commande `\footnote{`*texte de la note*`}`. Automatiquement, sur la page courante, il sera ajouté une note de bas de page, dont le texte sera celui que l'on retrouve dans la définition de la note.
Remarque. La numérotation est automatique.

3.8 Eléments flottants

Dans certains documents, on insère des tableaux, des images, des dessins,... mais la position qu'ils occupent dans le texte a peu d'importance. En effet, on y fait référence par un numéro (cf. 7.10). On les appelle des éléments flottants. LaTeX propose deux environnements différents pour positionner ces éléments. Ces deux environnements ont le même effet au fait près qu'ils identifient les éléments par des noms différents et des numérotations distinctes (voir la commande `\caption` plus bas).

`table` est libellé par **Tableau** (ou mot anglais) et
`figure` est libellé par **Figure** (ou mot anglais).

La commande pour ces environnements est
`\begin{` *name* `}[` *loc* `]` *body* `\end{` *name* `}`
où

name est le nom de l'environnement (`table` ou `figure`),

loc est une séquence d'au plus 4 caractères parmi `h t b p` suivant l'emplacement désiré (`h (here)` emplacement non-flottant, `t (top)` en haut d'une page, `b (bottom)` au bas de la page, `p (page of floats)` sur une page séparée ne contenant que des éléments flottants),

body est le texte (ou autre) à placer.

L'utilité de ces environnements est très simple. Supposons que l'on désire placer un tableau qui demande les trois-quarts d'une page. Si ce tableau est inséré sur une page qui contient une moitié de texte, il y aura un blanc d'une demi page. Avec les éléments flottants, le tableau sera placé au début de la page qui suit mais du texte sera inséré sur le demi page qui aurait été vide (pour autant qu'il y ait du texte qui suit le tableau).

Pour obtenir l'intitulé de l'environnement ainsi qu'un numéro, il faut utiliser la commande
`\caption[` *entree* `]` { *texte* }
qui provoque l'affichage du texte après l'intitulé et le numéro. Plusieurs commandes `\caption` peuvent apparaître dans un même environnement. Le texte *entree* sert d'entrée pour la liste des figures ou tableaux.

Deux commandes permettent de créer la liste des tableaux et la liste des figures, `\listoftables` et `\listoffigures` respectivement. Si la commande `\caption` ne contient pas d'entrée, c'est le texte qui sert d'entrée.

3.9 Les environnements prédéfinis

Nous avons déjà cité quelques environnements qui sont prédéfinis par LaTeX. Nous allons ici décrire leur utilité. Nous ne définirons cependant pas les environnements `list` et `trivlist`.

3.9.1 `center`, `flushleft` et `flushright`

L'environnement `center` est utilisé pour produire une ou plusieurs lignes de texte centré sur la page. La commande `\\` permet de passer à la ligne. Les paragraphes sont également permis mais ne sont pas indentés.

`flushleft` produit un texte aligné à gauche et non justifié (les lignes ne se terminent pas toutes à la marge de droite).

`flushright` produit un texte aligné à la marge de droite et non justifié (donc ne commence pas nécessairement à la marge de gauche).

Les commandes `\centering`, `\raggedleft` et `\raggedright` modifient les paramètres de mise en page de LaTeX et s'utilise dans un environnement pour produire un effet similaire aux trois environnements définis ci-dessus.

3.9.2 description, enumerate et itemize

LaTeX propose trois environnement pour créer des listes. Pour les trois, chaque nouvelle ligne débute par la commande `\item` qui accepte un argument optionnel. Par défaut, `itemize` commence chaque nouvelle ligne par un point ou un tiret (suivant la définition du style), `enumerate` par un nombre et `description` par l'argument optionnel du `\item`.

3.9.3 poetry

Cet environnement est similaire à `verse`. Une nouvelle strophe s'obtient en laissant une ligne vide. Une nouvelle ligne s'obtient par un double \.

3.9.4 quote, quotation et verse

LaTeX propose trois environnements pour les citations. Les marges gauche et droite sont déplacées de la même distance.

`quote` est utilisé pour de brèves citations, séparées par un ligne vide (dans le texte d'entrée). Il n'y a pas de retrait de paragraphe et un espace vertical supplémentaire est ajouté entre les paragraphes.

`quotation` est utilisé pour des citations de plus d'un paragraphe, séparés par une ligne vide également. Il y a un retrait de paragraphe et l'espace normal entre les paragraphes est utilisé.

`verse` est utilisé pour un poème. Les lignes d'une même strophe sont séparées par \\ et les strophes sont determinées par une ou plusieurs lignes vides.

3.9.5 tabbing

Cet environnement produit une séquence de lignes (en mode texte) avec un alignement basé sur des tabulations. Les tabulations sont numérotées 0, 1, 2,... Une tabulation est dite définie si on lui a assigné une position horizontale sur la page. La tabulation 0 est toujours définie par défaut (côté gauche de la page) et si une tabulation i est définie, les précédentes le sont également.

Le fonctionnement est basé sur les valeurs `next_tab_stop` et `left_margin_tab`. Initialement la valeur de `next_tab_stop` est 1 et celle de `left_margin_tab` est 0. La valeur de `next_tab_stop` est incrémentée par `\=` ou `\>` et réinitialisée par \\ ou `\kill`. Les commandes suivantes peuvent être utilisées dans l'environnement.

`\=` Si la valeur de `next_tab_stop` est i, alors la tabulation i est initialisée à cette position et la valeur de `next_tab_stop` est incrémentée.

`\>` si `next_tab_stop` vaut i, alors le texte qui suit débute à la tabulation i (même si elle est à gauche de la position où l'on se trouve) et la valeur de `next_tab_stop` est incrémentée.

\\ Débute une nouvelle ligne en initialisant `next_tab_stop` à `left_margin_tab`.

`\kill` Elimine la ligne courante en gardant les définitions des tabulations de cette ligne, initialise `next_tab_stop` à `left_margin_tab`.

`\+` Augmente d'une unité la valeur de `left_margin_tab`.

`\-` Diminue la valeur de `left_margin_tab`, qui doit être positive, d'une unité.

`\<` Diminue la valeur de `next_tab_stop` d'une unité. Cette commande ne peut être utilisée qu'en début de ligne pour annuler l'effet d'un `\+` pour cette ligne.

`\'` Place le texte juste à droite de la tabulation suivante ou contre la marge de gauche.

`` \` `` Place le texte contre la marge de droite. Il ne peut y avoir de `\>`, `\=` ou de \ commande après sur cette ligne.

`\pushtabs` Sauve les positions des tabulations pour les réutiliser lors de la commande `\poptabs` suivante.

`\poptabs` voir `\pushtabs`

`\a...` Les commandes `\=`, `\'` et `` \` `` produisent généralement des accents mais sont redéfinies dans un `tabbing`. Les commandes `\a=`, `\a'` et `` \a` `` remplacent ces commandes.

3.9.6 `tabular`

Pour mettre en forme un tableau avec LaTeX, il existe deux environnements: `tabular` et `array`. Ceux-ci s'utilisent de manière similaire. La différence entre les deux est simple: par défaut, le mode texte est utilisé pour les éléments des cellules dans un `tabular` alors que c'est le mode mathématique pour un `array`. On distingue les deux environnements essentiellement par l'utilité que l'on en fait: `array` est utilisé pour créer des matrices (ou tout autre tableau dont les éléments nécessitent le mode mathématique) et `tabular` pour créer des tableaux dont les éléments sont en mode texte. Un tableau est considéré par LaTeX comme un bloc de texte (`parbox`), c'est comme si c'est un grand caractère qui ne peut être coupé (donc doit être plus petit qu'une page).

Décrivons les notions qui sont toutes communes aux deux. Ce que nous écrivons pour `tabular` dans ces paragraphes reste valable pour `array`.

Ces environnements nécessitent un argument supplémentaire pour spécifier le nombre de colonnes. La syntaxe d'un tableau est
`\begin{tabular}[`*pos*`]{`*cols*`}` *lignes* `\end{array}`
où

pos est un argument optionnel qui spécifie la position en hauteur du tableau sur la ligne; par défaut, le tableau est centré, un `t` aligne le dessus avec la ligne et un `b`, le dessous.

cols précise le format des colonnes. Chaque caractère correspond à une colonne ou une séparation de colonne.

- **l** Une colonne où les éléments sont alignés à gauche.
- **r** Une colonne où les éléments sont alignés à droite.
- **c** Une colonne où les éléments sont centrés.
- **|** Une ligne verticale entre deux colonnes.
- **@{**_texte_**}** Insère le texte dans toutes les lignes de la colonne. Le texte est en mode texte ou mathématique suivant que l'on est dans un `tabular` ou un `array`. Cette commande annule l'espace normal entre les deux colonnes où elle se trouve.
- **p{**_larg_**}** produit une colonne de largeur *larg* où le texte est comme dans une commande `\parbox`.
- ***{**_nbr_**}{**_cols_**}** est équivalent à *nbr* copies de *cols* qui peut lui même contenir une *–expression.

lignes est une séquence de lignes séparées par `\\`. Chaque ligne est une séquence d'éléments séparés par & et il doit y avoir le même nombre d'éléments que de colonnes spécifiées par *cols*. Chaque élément est compilé comme s'il se trouvait entre accolades donc l'effet d'une macro se limite à cet élément. Les commandes suivantes peuvent apparaître dans les lignes.

- `\multicolum`{*num*}{*col*}{*texte*} fait de *texte* l'élément d'une cellule qui prend la place de *num* cellules dans la lignes où elle se trouve. Si *num*=1, alors la commande sert uniquement à changer la position de cet élément. *col* peut contenir exactement les mêmes caractères que *cols* mais ne peut contenir qu'une seule colonne.
- `\vline` Lorsqu'elle est utilisée à l'intérieur d'une colonne, elle produit une ligne verticale de la hauteur de la cellule.

Les instructions suivantes peuvent venir entre les colonnes pour produire des lignes horizontales.

- `\hline` produit une ligne horizontale sur toute la largeur du tableau.
- `\cline`{`i - j`} produit une ligne horizontale de la colonne `i` à la colonne `j` incluses.

La largeur des colonnes est automatiquement ajustée à la largeur du plus large élément de cette colonne. Il ne peut pas y avoir un & après le dernier élément d'une ligne. On peut inclure un tableau dans un tableau (récursivement).

3.9.7 verbatim

L'environnement `verbatim` restitue exactement le texte d'entrée en sortie. Il respecte tous les caractères (même réservés) ainsi que les espaces et les fins de ligne. La police utilisée est *typewriter*. Cet environnement commence une nouvelle ligne et passe à la ligne lorsqu'il se termine.

Il est possible de produire quelques caractères affichés exactement comme introduits mais qui ne se trouvent pas sur une ligne séparée. La commande à utiliser est `\verb` et sa syntaxe est un peu particulière. Cette commande prend un argument (le texte à afficher) délimité par une paire de caractères identiques (à l'exception de l'espace, de * et d'une lettre). Ainsi les commandes `\verb-$\\'e &-` et `\verb+$\\'e &+` produisent le même texte qui est `$\\'e &`.

Il y a également l'environnement `verbatim*` et la commande `\verb*`. Elles agissent exactement comme `verbatim` et `\verb` au fait près qu'un espace produit ␣. Ainsi `\verb*-$\\'e &-` produit `$\\'e␣&`.

Il y a trois restrictions à l'utilisation de cet environnement et de cette commande: (1) ils ne peuvent apparaître dans un argument d'une commande mais bien dans un autre environnement, (2) il ne peut y avoir aucun espace entre la commande `\verb` ou `\verb*` et son argument et (3) il ne peut y avoir d'espace entre `\end` et `{verbatim}`.

3.10 Des pages dans une page

LaTeX propose deux environnements qui permettent de créer un texte justifé mais de largeur inférieure à la largeur de la page ou de la colonne.

`\parbox[`*pos*`]{`*wdth*`}{`*texte*`}` produit un bloc de largeur *wdth* contenant le texte *texte*. Ce bloc est considéré comme un simple caractère. Sa position sur la ligne est spécifiée par *pos*. Par défaut, le bloc est centré, si *pos*=`t`, la ligne supérieure est alignée avec le texte et si *pos*=`b`, la ligne inférieure est alignée avec le texte.

L'environnement `\begin{minipage}[`*pos*`]{`*wdth*`}` *texte* `\end{minipage}` produit le même bloc de texte. La différence entre les deux est l'utilisation des notes de bas de page. Dans une *minipage*, les notes de bas de page apparaissent dans le bloc de texte produit par l'environnement. De plus, dans cet environnement, on peut utiliser tous les autres environnements comme `itemize`,... ce qui n'est pas le cas dans une commande `\parbox`.

3.11 Numéroter les pages

LaTeX numérote par défaut les pages suivant le style. Il est possible de supprimer les numéros ou de changer leur forme. Pour les supprimer, on consultera le guide de référence. Pour les numéroter différemment, on utilise la commande `\pagenumbering{`*style*`}` où *style* est un des mots `arabic` (nombres arabes), `roman` (nombres romains minuscules), `Roman` (nombres romains majuscules), `alph` (lettres minuscules) ou `Alph` (lettres majuscule). Par exemple, pour obtenir un document qui commence par la numérotation romaine pour l'introduction, la table des matière,... et qui se poursuit par la numérotation arabe, placez la commande `\pagenumbering{roman}` avant le texte et `\pagenumbering{arabic}` juste après la commande `\chapter` du premier chapitre.

4 Mises en forme

Nous entendons par mises en forme, les possibilités de disposer les caractères différemment sur une page et celles de changer la taille des caractères.

4.1 Taille des caractères

La taille des caractères est en premier lieu définie par les options spécifiées au début d'un document (voir *options* de la section 2). En plus de cela, il existe 10 tailles que l'on peut spécifier par les commandes du tableau 3. L'effet d'une telle commande est de changer la taille jusqu'à la fin de l'environnement courant (éventuellement jusqu'à la fin du document). Pour ne changer que la taille de quelques mots, on les place entre accolades avec la commande de taille au début.

4.2 Polices de caractères

La police par défaut de LaTeX est la police roman. D'autres polices sont disponibles pour mettre en évidence certaines parties de texte.

This is a bold type style.	{\bf This is a bold type style.}
This is an italic type style.	{\it This is an italic type style.}
This is an emphasized type style.	{\em This is an emphasized type style.}
This is a sans serif type style.	{\sf This is a sans serif type style.}
This is a slanted type style.	{\sl This is a slanted type style.}
THIS IS A SMALL CAPS TYPE STYLE.	{\sc This is a Small Caps type style.}
`This is a typewriter type style.`	{\tt This is a typewriter type style.}
This is a roman type style.	{\rm This is a roman type style.}

Dans le cas où la police italique (ou emphasized) ne porte que sur une partie d'un mot, il faut ajouter \/ avant de repasser en mode normal. Ainsi *bon*jour et *bon*jour sont différents et obtenus en tapant respectivement {\em bon}jour et {\em bon\/}jour.

Deux instructions du même changement de police ont un effet d'annulation. Ainsi l'instruction `{\em Un {\em texte accentué} dans un texte accentué}` produit le texte *Un* texte accentué *dans un texte accentué*.

Remarque. Pour obtenir un style d'écriture large et gras, il faut utiliser la combinaison de commandes `\large\bf` et non l'inverse.

5 Formules mathématiques

Des formules mathématiques peuvent être placées à l'intérieur du texte ou sur un ligne séparée. Plusieurs environnements sont disponibles pour introduire de telles formules.

On dit que LaTeX est en mode mathématique (à l'opposé du mode texte ou LR mode) s'il est à l'intérieur d'un environnement mathématique. Dans ce mode, il ignore les espaces introduits dans le texte. Les espaces sont cependant indispensables pour séparer les éléments, notamment les commandes. De même les espaces permettent d'aérer le texte d'entrée pour des raisons de clarté. Pour insérer des espaces dans le texte, on consultera la partie 5.6.

Les commandes que nous allons voir dans la suite requièrent le *mode mathématique*.

5.1 Les environnements mathématiques

Pour placer une formule dans un texte, on utilise l'environnement `math`. LaTeX propose deux raccourcis pour cet environnement (`\(...\)` et `$...$`) qui s'utilisent à la place de `\begin{math}...\end{math}`.

Pour créer une formule non numérotée sur une ligne séparée, on utilise l'environnement `displaymath` qui possède deux raccourcis: `\[...\]` et `$$...$$`.

Pour créer une formule numérotée sur une ligne séparée, on utilise l'environnement `equation` qui ne possède pas de raccourci. Nous verrons dans la partie 7.10 comment faire référence au numéro de la formule.

D'autres environnements mathématiques comme `array` seront vus dans la suite.

Remarque. La mise en forme des formules diffère suivant que la formule se trouve dans un texte ou sur une ligne séparée. Cependant il est possible de forcer le style que l'on désire (par exemple une mise en forme comme dans un texte pour une formule séparée). Le style correspondant à une formule dans un texte est `\textstyle` et celui d'une formule centrée `\displaystyle`. Il suffit, pour forcer un style, d'entourer la formule (sans la déclaration d'environnement) d'accolades et du nom de la commande au début. (Ex. `${\displaystyle...}$`)

5.2 Structures courantes

5.2.1 Indices et exposants

Un indice s'obtient à l'aide de _ et un exposant à l'aide de ^. Voici quelques exemples.

x_i	`x_i`	x^2	`x^2`
x_i^2	`x_i^2`	x_{n_i}	`x_{n_i}`

5.2.2 Fractions

La commande `\frac{`*num*`}{`*den*`}` produit une fraction de numérateur *num* et de dénominateur *den*.

5.2.3 Racines

La racine carrée s'obtient par `\sqrt{...}` et la racine $n^{\text{ième}}$ par `\sqrt[n]{...}`.

5.2.4 Points de suspension

LaTeX propose quatre types de points de suspension résumés dans le tableau suivant:

$\ldots$	`\ldots`	$\cdots$	`\cdots`
$\vdots$	`\vdots`	$\ddots$	`\ddots`

La commande `\ldots` peut s'utiliser également en mode texte. La distinction entre `\ldots` et `\cdots` est la hauteur sur la ligne. Par exemple on utilise `\cdots` dans $x_1 + \cdots + x_n$ tandis que l'on utilise `\ldots` dans $x_1, \ldots, x_n$.

5.2.5 Symboles spéciaux

En mathématique, on utilise de nombreux symboles tels que les relations ensemblistes, les sommes, les intégrales, les limites, les lettres grecques, les lettres caligraphiques, les flèches,... Ces symboles sont repris dans les tableaux en fin de document (regroupés pour d'évidentes raisons de recherche).

5.2.6 Symboles à taille variable

Certains symboles, tels que somme et intégrale, changent de taille suivant les conditions. Un tableau contenant ces symboles se trouve en fin de document. Ces symboles peuvent posséder des indices et exposants comme pour le signe d'intégration où cela représente les bornes. Par exemple l'intégrale de zéro à l'infini de la fonction f s'obtient par le texte `\int_{0}^{\infty}f(x)\,dx` et produit

$$\int_{0}^{\infty} f(x)\,dx$$

Le but du `\,` est de fournir un petit espace entre la fonction et le `dx`.

5.3 Tableaux mathématiques

Un tableau en mathématique s'obtient par l'environnement `array`. Nous avons vu au point 3.9.6 son utilisation qui est identique à celle de `tabular`. Dans un `array`, il ne peut y avoir de `\\` sur la dernière ligne sinon cela produit un espace supplémentaire dans le texte et aucun avertissement n'est donné.

5.4 Aligner les formules

Certaines formules sont trop longues et doivent être écrites sur plusieurs lignes. On peut aussi désirer écrire une suite d'équations alignées sur plusieurs lignes... LaTeX propose deux environnements pour réaliser ces formules: `eqnarray` et `eqnarray*`.

L'environnement `eqnarray` peut être vu comme un tableau (centré sur la page) à trois colonnes dont la première et la troisième sont en mode mathématique et celle du milieu en mode texte. Les lignes sont séparées par une \\ et les colonnes par &. En plus, chaque ligne porte un numéro d'équation. On peut le supprimer en insérant sur la ligne la commande `\nonumber`.

L'environnement `eqnarray*` est identique à `eqnarray` sauf qu'il n'insère pas de numéro.

La commande `\lefteqn{...}` permet de positionner totalement à gauche la première ligne dans un des deux environnements. LaTeX pense que la longueur de la formule est alors 0.

5.5 Placer l'un au dessus de l'autre

Les tableaux peuvent être utilisés pour positionner des éléments les uns au dessus des autres mais ne peuvent résoudre tous les problèmes.

5.5.1 Les accents mathématiques

Dans le texte, les accents sont obtenus à l'aide des commandes du tableau 1. En mathématique, on ne peut les utiliser mais il existe d'autres commandes regroupées dans le tableau 2.

Les accents `\hat` ($\hat{}$) et `\tilde` ($\tilde{}$) possèdent une version qui s'élargit en fonction de son argument. Ces accents sont obtenus par les commandes `\widehat` et `\widetilde`. Ainsi on peut avoir $\widehat{1-x^2}$.

Les lettres i et j possèdent toujours un point. En mathématique, il existe deux commandes produisant les lettres sans les points pour pouvoir par exemple les utiliser avec les accents. On a `\imath` ($\imath$) et `\jmath` ($\jmath$).

5.5.2 Sou– et sur– ligner

La commande de soulignement est `\underline` et celle de surlignement est `\overline`. La commande de soulignement peut aussi être utilisée en mode texte.

5.5.3 Accolades horizontales

Une accolade peut être placée au-dessus ou au-dessous d'éléments par les commandes `\overbrace` et `\underbrace`. Un texte peut être ajouté au-dessus d'une accolade qui se trouve au-dessus d'une formule par `^` et inversement, par `_` pour l'autre cas.

`$\overbrace{a+b+c}$`	produit	$\overbrace{a+b+c}$
`$\overbrace{a+b+c}^d$`	produit	$\overbrace{a+b+c}^d$

5.5.4 *stacking symbol*

La commande `\stackrel` place un élément au-dessus d'un autre. Le symbole de définition $\stackrel{\triangle}{=}$ est obtenu par `\stackrel{\triangle}{=}`.

5.6 Espaces en mathématique

Comme dit plus haut, LaTeX ignore complètement les espaces dans les formules mathématiques. Plus exactement, il adapte lui-même les espaces. Dans certains cas, il est utile d'augmenter l'espace et dans d'autre de le diminuer. Pour cela, LaTeX offre quatres commandes:

`\,`	petit espace	`\:`	moyen espace
`\!`	petit espace négatif	`\;`	large espace

Seule `\,` peut aussi être utilisé sans le mode mathématique. Il faut utiliser ces commandes mais il ne faut pas en abuser car LaTeX fonctionne généralement bien. Il faut ajouter un petit espace avant le dx dans une intégrale, juste après une fraction et une racine. Un espace négatif est utilisé pour rapprocher les signes d'intégration d'une intégrale double ainsi qu'après la barre de fraction / d'un quotient.

5.7 Les délimiteurs

Un délimiteur est une barre, une accolade qui se place devant ou derrière un élément en s'ajustant à sa taille. Par exemple, devant un système d'équation, on place une grande accolade. LaTeX considère tous les symboles du tableau 11 comme des délimiteurs et apparaissent avec la même taille. Pour obtenir des délimiteurs qui s'ajustent à la taille des éléments qu'ils délimitent, on utilise les commandes `\left` et `\right` placées juste devant le délimiteur. Ces commandes vont toujours par paire (un gauche et un droit) et s'ajustent à la taille de ce qui se trouve entre les deux. Le délimiteur gauche peut ne pas être le même que celui de droite. Il est possible d'obtenir un délimiteur invisible simplement en plaçant un point (.) à la place du délimiteur.

6 Créer la bibliographie

Une partie importante d'un document est sa bibliographie. Elle doit être rigoureuse et claire. LaTeX vous aide en proposant un environnement spécialement conçu appelé `thebibliography`. La syntaxe en est la suivante
`\begin{thebibliography}`{*max-larg*} *entree* `\end{thebibliography}`
où

max-larg représente la largeur maximale des étiquettes des entrées de la bibliographie. Ce paramètre contrôle la mise en page.

entree est une liste d'entrée commançant par
`\bibitem[`*etiq*`]`{*citeEtiq*}
qui génère une entrée dont l'étiquette est *etiq*. Si ce paramètre est manquant, c'est le numéro de l'entrée qui est affiché. Le second paramètre est l'étiquette pour les références par la commande `\cite` (voir plus bas).

La commande `\cite[`*texte*`]`{*citeEtiqListe*} produit l'affichage des étiquettes (définies par la commande `\bibitem`) correspondant aux étiquettes de référence de la liste. Si *texte* est présent, il est ajouté comme commentaire à cette référence.

7 Remarques et astuces

7.1 Sectionner le texte source

Les fichiers d'entrée de LaTeX sont créés à l'aide d'un éditeur de textes (tel que "edit" sous DOS, "emacs" sous UNIX...). Losqu'un tel fichier devient long (plus de 1000 lignes), il n'est pas facile de voyager au sein de celui-ci. Pour éviter les longs fichiers, LaTeX propose deux commandes d'inclusion de fichier dans un autre.

`\input[` *file* `]` importe le texte qui se trouve dans le fichier *file* dans le fichier qui contient la commande, exactement comme si ce texte faisait partie de ce document.

`\include[` *file* `]` et `\includeonly[` *file-list* `]` sont deux commandes qui vont de paire. La première est celle qui produit l'inclusion d'un fichier et la deuxième conditionne les inclusions par la première. Si *file* est un des noms de *file-list*, la commande `\include[` *file* `]` est équivalente à `\clearpage` `\input[` *file* `]` `\clearpage`, sinon elle est équivalente à `\clearpage`. Si la commande `\includeonly` n'est pas présente, la première équivalence est d'application.

La commande `\includeonly` ne peut apparaître que dans le *preambule* et `\include` ne peut pas y apparaître ni dans un autre fichier lu par la même commande.

Une bonne méthode, à mon avis, pour sectionner un document est la suivante. Placer chaque chapitre dans un fichier. De même pour la bibliographie et l'introduction. Créer un fichier principal qui contiendra pour seules lignes les lignes données dans la partie 2. Le texte de ce document maître ne devra contenir que des `include` et éventuellement des commandes pour la tables des matières, la numérotation des pages,.... Le *preambule* contiendra la commande `includeonly` avec le nom du fichier qui contient le chapitre sur lequel on travaille (en général, on ne travaille que sur un chapitre). Cette subdivision permet d'économiser beaucoup de temps de compilation et conserve toujours les références croisées.

7.2 Les tirets

Les mots composés sont séparés par un tiret généralement assez court. Il est possible de créer des tirets plus long en doublant ou triplant le caractère. Ainsi un tiret donne - et trois tirets donnent —.

7.3 Opérateurs unaires ou binaires?

Un caractère - ou + qui débute une formule est assimilé à un opérateur unaire qui précéde sans espace l'élément qui suit. Ainsi `$+x$` donne $+x$. Cependant, lorsqu'une formule est divisée, le symbole doit être interprété comme un opérateur binaire et il faut le préciser au moyen d'un premier argument invisible obtenu par un `\mbox` avec un argument vide. Donc `$\mbox{}+x$` donne $\mbox{}+x$.

7.4 Obtenir un *prime*

Le caractère mathématique *prime* ($'$) souvent utilisé s'obtient par la commande `^\prime` mais peut également s'obtenir par l'apostrophe. Ainsi `x'` est équivalent à `x^\prime`.

7.5 Espaces et points

Généralement l'espace entre un point terminant une phrase et le début de la phrase suivante est plus long que l'espace séparant deux mots. Pour déterminer si un point termine une phrase ou non, LaTeX utilise une règle très simple: un point (suivit d'un espace) termine une phrase sauf s'il suit une lettre majuscule. En générale, cela fonctionne très bien mais il peut arriver qu'il y ait confusion comme dans l'abréviation "etc." qui constitue une des rares exceptions. Pour indiquer à LaTeX qu'un point ne termine pas une phrase, il suffit de placer `\␣` directement après le point (sans espace entre le point et le \). On peut laisser plusieurs espaces après si on le désire mais aucun avant. Dans le cas inverse, on indique à LaTeX qu'un point qui suit une majuscule termine une phrase par un `\@` juste avant le point.

Si un point précède directement une parenthèse fermante, LaTeX ajoute également un espace supplémentaire après la parenthèse. Comme pour le point, on indique par un \ suvit d'un espace directement après la parenthèse que l'espace doit être normal.

La même règle doit également être utilisée pour le point d'interrogation (?), pour le point d'exclamation (!) et pour les deux points (:).

7.6 Points de suspension

Les points de suspension ne peuvent pas être obtenus en plaçant trois points successifs car les espaces entre les points ne sont pas corrects dans ce cas. Pour obtenir "…", on utilise la commande `\ldots` (qui peut s'utiliser en mode texte comme en mode mathématique).

7.7 Crochets et arguments optionnels

La commande `\item` peut prendre un argument optionnel placé entre crochets. Si comme argument, on désire mettre un texte entre crochets ou si on désire commencer le texte par un crochet, il peut y avoir confusion pour LaTeX. On y remédie comme suit: si l'argument optionnel comporte des crochets, on place l'argument entre des accolades et les accolades entre crochets, si le texte commence par un crochet, on place ce texte entre accolades.

7.8 Coupures indésirables

La coupure des lignes est quasi entièrement automatique et inchangeable à l'intérieur d'un paragraphe. Cependant une fin de ligne dans le texte peut devenir indésirable comme lorsque l'on fait référence à un numéro de chapitre, une page, ... Si on écrit dans le texte d'entrée `voir page 2`, rien n'empèche LaTeX de terminer la ligne après le mot `page`. Pour empêcher cette coupure, on utilise le tilde (`~`) qui produit un espace mais relie les deux mots qu'il sépare pour n'en former qu'un. Ainsi on écrira `voir page~2`.

Une autre possibilité pour empêcher la coupure d'une suite de mots est la commande `\mbox{...}`. Le texte se trouvant entre les crochets ne sera jamais coupé.

Il ne faut pas abuser dans l'utilisation de ces commandes sous peine de provoquer de nombreux avertissement de compilation (`overfull hbox`*). En effet, LaTeX n'est plus libre de couper les mots où il le désire.*

7.9 Retrait de paragraphe

Dans la littérature anglaise, le premier paragraphe d'un chapitre, d'une section,... n'est pas indenté. En français, le premier paragraphe doit être indenté exactement comme les autres. La commande `\indent` produit un espace équivalent à celui d'un retrait de paragraphe. Pour créer un document français lorsqu'on ne possède pas de style français, il faut ajouter cette commande avant chaque premier paragraphe. *Je ne sais pas pourquoi, mais sur la version que j'utilise, je suis obligé de doubler la commande ou d'ajouter un argument nul devant. J'ai pour cela créer une macro* `\Ret` *qui produit cet espace. Si un style français est disponible, il me suffit de redéfinir cette macro comme nulle et le document reste valide.*

7.10 Références croisées

Nous appelons *référence croisée* la possibilité offerte par LaTeX d'insérer automatiquement un numéro de théorème, de page, de chapitre,... qui est reconnu par un nom. Expliquons cela sur un exemple.

Pour obtenir le titre de cette partie, nous avons tapé
`\subsection{R\'ef\'er\'ences crois\'ees}`
et nous y avons ajouté `\label{crossref}`. Cet ajout ne produit aucun texte dans le document mais pose une étiquette (ici `crossref`). La commande `\ref{crossref}` placée n'importe où dans le document (aussi bien avant que l'étiquette soit définie qu'après) sauf dans un environnement tel que `verbatim`, produit le texte *7.10*. Ce texte correspond au numéro de la sous section dans ce cas.

Une étiquette peut être utilisée dans tout environnement portant un numéro (les théorèmes, les chapitres, les sections,...). Lorsqu'elle est utilisée avec la commande `\caption` dans une figure ou un tableau, la commande `\label` doit se trouver après le `\caption`.

En rapport avec `\label`, il y a aussi la commande `\pageref` qui donne le numéro de la page sur laquelle se trouve l'étiquette (ici, 15).

7.11 Style mathématique

Il est important de remarquer que dans une formule mathématique, la police de caractères n'est pas celle par défaut de LaTeX (càd. la police roman). Les formules mathématiques utilisent la police *emphasized*. Cependant il ne faut pas taper `$efficace$` au lieu de `{\em efficace}` car le premier donne $efficace$: et le second *efficace*. L'espacement des lettres n'est pas le même.

Vu cette remarque, lorsque dans un texte, on désire faire référence à, par exemple, une fonction qui apparaît dans une formule, soit f, il est important d'entourer la lettre `f` de `$`. En effet, si on ne le fait pas, la lettre apparaîtra en style roman, donc f, tandis qu'avec des `$`, on aura f.

Tableau 1: Les accents

ò	`\`{o}`	ó	`\'{o}`	ô	`\^{o}`	ö	`\"{o}`	õ	`\~{o}`
ō	`\={o}`	ȯ	`\.{o}`	ŏ	`\u{o}`	ǒ	`\v{o}`	ő	`\H{o}`
o͡o	`\t{oo}`	o̧	`\c{o}`	ọ	`\d{o}`	o̲	`\b{o}`		

Tableau 2: Les accents mathématiques

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

Tableau 3: Taille des caractères

Bon	\tiny	Bon	\scriptsize	Bon	\footnotesize	Bon	\small
Bon	\normalsize	Bon	\large	Bon	\Large	Bon	\LARGE
Bon	\huge	Bon	\Huge				

Tableau 4: Lettres grecques

α	\alpha	β	\beta	γ	\gamma	δ	\delta	ϵ	\epsilon
ε	\varepsilon	ζ	\zeta	η	\eta	θ	\theta	ϑ	\vartheta
ι	\iota	κ	\kappa	λ	\lambda	μ	\mu	ν	\nu
ξ	\xi	π	\pi	ϖ	\varpi	ρ	\rho	ϱ	\varrho
σ	\sigma	ς	\varsigma	τ	\tau	υ	\upsilon	ϕ	\phi
φ	\varphi	χ	\chi	ψ	\psi	ω	\omega	Γ	\Gamma
Δ	\Delta	Θ	\Theta	Λ	\Lambda	Ξ	\Xi	Π	\Pi
Σ	\Sigma	Υ	\Upsilon	Φ	\Phi	Ψ	\Psi	Ω	\Omega

Tableau 5: Symboles mathématiques

$\pm$	\pm	$\mp$	\mp	$\times$	\times	$\div$	\div
$\ast$	\ast	$\star$	\star	$\circ$	\circ	$\bullet$	\bullet
$\cdot$	\cdot	$\cap$	\cap	$\cup$	\cup	$\uplus$	\uplus
$\sqcap$	\sqcap	$\sqcup$	\sqcup	$\vee$	\vee	$\wedge$	\wedge
$\setminus$	\setminus	$\wr$	\wr	$\diamond$	\diamond	$\bigtriangledown$	\bigtriangledown
$\lhd$	\lhd	$\bigtriangleup$	\bigtriangleup	$\rhd$	\rhd	$\triangleleft$	\triangleleft
$\unlhd$	\unlhd	$\triangleright$	\triangleright	$\unrhd$	\unrhd	$\oplus$	\oplus
$\ominus$	\ominus	$\otimes$	\otimes	$\oslash$	\oslash	$\odot$	\odot
$\bigcirc$	\bigcirc	$\dagger$	\dagger	$\ddagger$	\ddagger	$\amalg$	\amalg

Tableau 6: Symboles de relation

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

La négation des symboles s'obtient en ajoutant `\not` devant l'expression de ceux-ci. Par exemple, $\not<$ et $\not\leq$ s'obtiennent respectivement en tapant `\not<` et `\not\leq`.

Tableau 7: Symboles à taille variable

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

Tableau 8: Flèches

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

Tableau 9: Divers symboles

<table>
<tr><td>$\aleph$</td><td>\aleph</td><td>$\hbar$</td><td>\hbar</td><td>$\imath$</td><td>\imath</td><td>$\jmath$</td><td>\jmath</td><td>ℓ</td><td>\ell</td></tr>
<tr><td>$\wp$</td><td>\wp</td><td>$\Re$</td><td>\Re</td><td>$\Im$</td><td>\Im</td><td>$\mho$</td><td>\mho</td><td>$\prime$</td><td>\prime</td></tr>
<tr><td>$\emptyset$</td><td>\emptyset</td><td>∇</td><td>\nabla</td><td>$\surd$</td><td>\surd</td><td>$\top$</td><td>\top</td><td>$\bot$</td><td>\bat</td></tr>
<tr><td>$\Vert$</td><td>\|</td><td>$\angle$</td><td>\angle</td><td>$\forall$</td><td>\forall</td><td>$\exists$</td><td>\exists</td><td>$\neg$</td><td>\nag</td></tr>
<tr><td>$\flat$</td><td>\flat</td><td>$\natural$</td><td>\natural</td><td>$\sharp$</td><td>\sharp</td><td>$\backslash$</td><td>\backslash</td><td>∂</td><td>\partial</td></tr>
<tr><td>∞</td><td>\infty</td><td>$\Box$</td><td>\Box</td><td>$\Diamond$</td><td>\Diamond</td><td>$\triangle$</td><td>\triangle</td><td>$\clubsuit$</td><td>\clubsuit</td></tr>
<tr><td>$\diamondsuit$</td><td>\diamondsuit</td><td>$\heartsuit$</td><td>\heartsuit</td><td>$\spadesuit$</td><td>\spadesuit</td><td></td><td></td><td></td><td></td></tr>
</table>

Tableau 10: Fonctions mathématiques

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

Tableau 11: Délimiteurs

<table>
<tr><td>(</td><td>(</td><td>)</td><td>)</td><td>[</td><td>[</td><td>]</td><td>]</td><td>$\{$</td><td>\{</td></tr>
<tr><td>$\}$</td><td>\}</td><td>$\lfloor$</td><td>\lfloor</td><td>$\rfloor$</td><td>\rfloor</td><td>$\lceil$</td><td>\lceil</td><td>$\langle$</td><td>\langle</td></tr>
<tr><td>$\rangle$</td><td>\rangle</td><td>/</td><td>/</td><td>$\vert$</td><td>|</td><td>$\Vert$</td><td>\|</td><td>$\uparrow$</td><td>\uparrow</td></tr>
<tr><td>$\downarrow$</td><td>\downarrow</td><td>$\updownarrow$</td><td>\updownarrow</td><td>$\Uparrow$</td><td>\Uparrow</td><td>$\Downarrow$</td><td>\Downarrow</td><td>$\Updownarrow$</td><td>\Updownarrow</td></tr>
</table>

Table des matières

1 Fonctionnement **1**

2 Nouveau document **1**

2.1 La syntaxe LaTeX . . . 2
2.2 Commande . . . 2
2.3 Environnement . . . 2
2.4 Théorème . . . 3
2.5 Fichier de style . . . 3

3 Mise en page **3**

3.1 Paragraphe . . . 3
3.2 Coupures des lignes et pages . . . 3
3.3 Les espaces . . . 4
3.4 Les espaces entre paragraphes et entre lignes . . . 5
3.5 Chapitres, sections, ... et annexes . . . 5
3.6 Table des matières . . . 5
3.7 Notes de bas de pages . . . 5
3.8 Eléments flottants . . . 6
3.9 Les environnements prédéfinis . . . 6
3.9.1 `center`, `flushleft` et `flushright` . . . 6
3.9.2 `description`, `enumerate` et `itemize` . . . 7
3.9.3 `poetry` . . . 7
3.9.4 `quote`, `quotation` et `verse` . . . 7
3.9.5 `tabbing` . . . 7
3.9.6 `tabular` . . . 8
3.9.7 `verbatim` . . . 9
3.10 Des pages dans une page . . . 9
3.11 Numéroter les pages . . . 9

4 Mises en forme **9**

4.1 Taille des caractères . . . 9
4.2 Polices de caractères . . . 10

5 Formules mathématiques **10**

5.1 Les environnements mathématiques . . . 10
5.2 Structures courantes . . . 11
5.2.1 Indices et exposants . . . 11
5.2.2 Fractions . . . 11
5.2.3 Racines . . . 11
5.2.4 Points de suspension . . . 11
5.2.5 Symboles spéciaux . . . 11
5.2.6 Symboles à taille variable . . . 11
5.3 Tableaux mathématiques . . . 11
5.4 Aligner les formules . . . 11
5.5 Placer l'un au dessus de l'autre . . . 12
5.5.1 Les accents mathématiques . . . 12

5.5.2 Sou– et sur– ligner . . . 12
5.5.3 Accolades horizontales . . . 12
5.5.4 *stacking symbol* . . . 12
5.6 Espaces en mathématique . . . 12
5.7 Les délimiteurs . . . 13

6 Créer la bibliographie **13**

7 Remarques et astuces **13**
7.1 Sectionner le texte source . . . 13
7.2 Les tirets . . . 14
7.3 Opérateurs unaires ou binaires? . . . 14
7.4 Obtenir un *prime* . . . 14
7.5 Espaces et points . . . 14
7.6 Points de suspension . . . 14
7.7 Crochets et arguments optionnels . . . 14
7.8 Coupures indésirables . . . 15
7.9 Retrait de paragraphe . . . 15
7.10 Références croisées . . . 15
7.11 Style mathématique . . . 15