

INITIATION À L^AT_EX

Table des matières

I	Les outils pour faire un bon rapport en L^AT_EX	2
1	Introduction	2
2	Installation et logiciels	2
3	Concepts	2
3.1	Classe	3
3.2	Packages	3
3.3	Préambule et corps du document	4
3.4	Fichiers générés	4
3.5	Commandes et environnements	4
4	Les commandes à connaître	5
4.1	Structure	5
4.2	Texte	6
4.3	Longueurs	6
4.4	Compteurs	7
4.5	Tableaux	7
4.6	Listes	8
4.7	Et les maths!	8
4.7.1	Environnements	8
4.7.2	Délimiteurs	9
4.7.3	Lettres grecques	9
4.7.4	Fontes mathématiques	9
4.7.5	Commandes	9
4.7.6	Matrices	9
4.7.7	Equations et références	10
4.7.8	Divers	11
4.8	Quelques commandes supplémentaires de mise en page	11
4.9	Figures	12
5	Liens et littérature	12
II	Supplément : BEAMER	13
1	Installation	13
2	Préambule	13
3	Quelques commandes utiles pour le rendu	14
4	Diapos	14

Première partie

Les outils pour faire un bon rapport en L^AT_EX

1 Introduction

L^AT_EX (et plus précisément L^AT_EX 2_ε) est un outil très puissant de traitement de texte scientifique. Il permet de réaliser des rapports, articles, livres, présentations et même des partitions de musique, voire bien plus encore. Il descend du T_EX développé initialement par Knuth à la fin des années 70.

Il peut être considéré comme un langage de programmation semi-interprété dans la mesure où, contrairement à Microsoft Office Word, la visualisation ne correspond à ce qui est tapé dans l'éditeur. Il existe en effet un grand nombre de commandes pour générer le rendu souhaité après compilation, mais la mise en page reste la chasse gardée du compilateur. Il est difficile (et déconseillé) de s'opposer à L^AT_EX en matière d'espaces (qu'ils soient horizontaux ou verticaux), de position des objets, . . . Le compilateur gère tous les paramètres de mise en page, la numérotation des chapitres, des paragraphes et des pages, les références, . . .

On peut distinguer deux niveaux d'utilisation de L^AT_EX : le premier est l'utilisateur de base, pour qui L^AT_EX est un outil en soi, qu'il finit par utiliser même en dehors du contexte scientifique ; le second est celui pour qui L^AT_EX devient un jeu et un objet de développement. Car on peut « coder » du L^AT_EX, et le manipuler à sa guise, ce qui requiert quelques connaissances et expériences.

2 Installation et logiciels

Pour commencer, il faut installer une distribution L^AT_EX (ici MiKTeX 2.8, qui est recommandée). Sous Windows, en trois clics sur le web, le tour est joué. A noter pour ceux qui auront remarqué la version de PDF_TE_X que chaque évolution de L^AT_EX lui fait gagner une décimale du nombre π .

Une fois le moteur installé, il reste la carrosserie. Il est certes possible de taper son texte dans n'importe quel éditeur de texte et sous quelque système d'exploitation que ce soit, puis de lancer le compilateur L^AT_EX en ligne de commande, mais pour ceux qui sont habitués aux interfaces graphiques, il existe plusieurs logiciels pour mâcher le travail. Du plus basique au plus évolué :

- livré par défaut avec MiKTeX, TeXworks ;
- un bon outil : TeXnicCenter (Windows), et son équivalent Linux : Kile ;
- à peu près équivalents sous Windows : WinShell ;
- ce qui se fait peut-être de mieux : WinEdt (mais payant, . . .)

3 Concepts

Une fois l'installation des composants terminés et avant de se lancer à corps perdu dans la rédaction du prochain compte-rendu de TP, il faut comprendre les concepts inhérents à L^AT_EX.

3.1 Classe

En premier lieu, la classe de document : c'est le style général de votre document. Il en existe autant que d'utilisateurs, puisque chacun peut créer sa propre classe (fichier `.cls`). Mais les plus courants sont :

- **article** : la classe de base pour faire des documents courts, comme des articles.
- **report** : la classe intermédiaire, qui, comme son nom l'indique, est idéale pour écrire un rapport, qui propose plus de hiérarchie dans les paragraphes (chapitre, partie).
- **book** : une classe plus évoluée avec une hiérarchie importante, et des options plus nombreuses quant à la disposition générale des pages.

Une fois votre classe choisie en fonction des propriétés du document à rédiger, vous pouvez commencer votre fichier `.tex` par :

```
\documentclass[opt]{classe}
```

Parmi les options `opt` possibles :

- **a4paper** : format du papier ;
- **11pt** : taille de la police (autres possibilités : 10pt, 12pt) ;
- **landscape** : mode paysage ;
- **draft** : option « brouillon », qui fait apparaître les problèmes de surcharge ;
- ...

Chaque classe a ses options par défaut. Pour imposer ses options, il suffit de les déclarer entre [] au début du document.

3.2 Packages

Pour pouvoir disposer de commandes spécifiques dans le domaines des symboles, de la gestion des figures, on fait appel à des *packages*, que l'on déclare à l'aide de la commande :

```
\usepackage{nom du package}
```

Les packages les plus couramment utilisés sont les suivants :

- **inputenc** avec option **latin1**, **fontenc** avec option **T1**, et **babel** avec option **frenchb** : permet d'écrire directement en français avec les accents, les cédilles, les césures selon les règles de la grammaire française ;
- **amsmath**, **amssymb** et **amsfonts** : packages spécifiques pour les maths, rendant disponibles une grande quantité de symboles de mathématiques, de structures d'équations, ... (le fichier `amslatex.pdf` est à lire absolument).
- **geometry** avec option **scale=r**, où r est un réel de $]0, 1[$ qui représente le rapport entre la taille de la zone de texte et la taille du papier. Par défaut, la mise en page d'un document L^AT_EX est celle d'une lettre américaine, *i.e.* une zone de texte très petite (faire l'essai). Le package **geometry** propose des proportions harmonieuses automatisées. Il y a bien sûr possibilité de contrôler manuellement les marges (commandes `\textheight`, `\hoffset`, ... mais le principe est déconseillé). Il existe d'autres packages qui traitent ce problème comme **layaureo**, ...
- **fancyhdr** : pour la gestion des en-têtes et des pieds-de-page.
- **graphicx** : pour la gestion des images insérées.
- **float** : pour forcer l'insertion des objets flottants.
- **xcolor** : pour un choix plus large de couleurs.

- `hyperref` : pour créer des liens hypertextes dans les documents.
- `refcheck` : affiche sur le document tous les labels associés à aux équations, paragraphes, ... pour faciliter le traitement des références dans la rédaction.
- `TikZ` : s'il fallait démontrer la puissance de L^AT_EX en un package, pour faire des dessins, des schémas, ...
- et des centaines d'autres packages.

Pour installer un package :

1. le télécharger (voir le § **Liens**)
2. placer l'archive dans le répertoire `texmf\tex\latex`
3. soit l'archive comprend le `.sty` (fichier de style, code du package), soit elle comprend un `.ins` (fichier d'installation). Dans le premier cas, il n'y a rien à faire. Dans le second, il faut lancer le compilateur L^AT_EX sur le fichier d'installation.
4. mettre à jour la base de données : sous Linux, il suffit de taper la commande `texhash`; sous Windows avec une distribution MiKTeX, ouvrir la fenêtre **Settings** et cliquer sur **Refresh FNDB**.

3.3 Préambule et corps du document

Un document L^AT_EX se décompose comme suit : la déclaration de la classe, puis le chargement des packages, des commandes et des environnements, puis vient le corps du texte compris entre `\begin{document}` et `\end{document}`. Tout ce qui suit n'apparaîtra pas dans le document de visualisation. Tout ce qui précède `\begin{document}` est appelé *préambule*.

3.4 Fichiers générés

Lors de la compilation du document est généré un fichier de visualisation `.dvi`, qui peut ensuite être converti en fichier postscript (`.ps`) ou en document portable (`.pdf`).

Mais de nombreux autres fichiers sont générés automatiquement : un fichier `.log` qui regroupe les sorties lors de la compilation, un fichier `.aux` pour les références croisées, un fichier `.toc` lorsqu'une table des matières est insérée dans le document, un fichier `.bbl` en cas d'existence d'une bibliographie, ...

3.5 Commandes et environnements

Ce qui fait la différence entre L^AT_EX et un éditeur de texte quelconque, outre le fait que la mise en page est gérée entièrement par le compilateur, c'est le fait de disposer d'un certain nombre de commandes qui produisent des symboles, des cadres, des listes, ...

Il existe un nombre important de **commandes** prédéfinies (de base ou dans les packages), mais on peut en définir de nouvelles à l'aide de la macro :

```
\newcommand{\nom}[nb arg]{déf commande}.
```

Cela crée un alias, de sorte qu'à chaque fois que vous taper `\nom[arg opt]{arg oblig}` est compilé en réalité `déf commande` (noter le `\` devant le nom de la commande). Par exemple, pour le symbole \mathbb{R} , plutôt que de taper `\mathbb{R}` à chaque fois, on définit en préambule `\newcommand{\R}{\mathbb{R}}`

ce qui représente un gain de temps non négligeable. Le nom à donner à la commande est libre, à quelques restrictions près sur certains symboles à ne pas utiliser ($\backslash \$ \sim _ \wedge \& \% \# \{ \}$).

On peut donner des arguments à la commande (optionnel donc entre $[\]$). Dans la définition de la commande, on fait appel aux arguments par $\#1, \#2, \dots$. Par exemple, si l'on veut rendre modulable les normes dans les espaces de Sobolev, on peut définir la macro :

```
\newcommand{\nhs}[2]{\left\|#2\right\|_{H^{\#1}}}
```

On y fait appel comme suit : $\$ \backslash nhs\{s\}\{f\}$, ce qui donne $\|f\|_{H^s}$. Lorsqu'une commande existe déjà mais que l'on veut changer sa définition, on utilise `\renewcommand`.

Un **environnement** est une structure plus riche. La commande de déclaration est la suivante :

```
\newenvironment{nom}[nb arg]{déf début environnement}{déf fin environnement}
```

L'appel se fait grâce à `\begin{nom} ... \end{nom}`. Par exemple, la déclaration suivante :

```
\newenvironment{Rq}{\noindent\textbf{Remarque}\quad}{~\hfill $\diamond$\medskip}
```

permet de faire des remarques (`\begin{Rq} texte \end{Rq}`), comme par exemple :

Remarque Ceci est une remarque. Tout est prédéfini et prêt à l'emploi. ◇

4 Les commandes à connaître

4.1 Structure

Pour structurer un document, on dispose d'une certaine hiérarchie. Par ordre d'importance, donnant lieu à numérotation et à entrée dans la table des matières :

- `\part`
- `\chapter` (non disponible dans la classe `article`)
- `\section` (numérotation liée à celle des chapitres)
- `\subsection` (idem)
- `\subsubsection` (ne donne plus lieu à une numérotation dans la classe `book`)

Il existe également des commandes `\paragraph` et `\subparagraph`, non numérotées. Pour disposer de la même taille de titre mais sans la numérotation, on ajoute une étoile à la commande (ex : `\section*{Titre}`). Enfin, la commande `\appendix` permet de changer le style de numérotation des chapitres ou des sections dans les annexes pour les distinguer du corps de texte principal.

Une commande utile dans les documents longs est `\tableofcontents` qui, comme son nom l'indique, génère la **table des matières** à l'endroit où est fait appel à la commande. Le package `minitoc` permet de générer des tables des matières partielles au début de chaque chapitre. Il est possible de gérer la profondeur de la table (y faire apparaître jusqu'aux sections, sous-sections, ...).

Pour faire une **page de garde**, il est nécessaire de préciser dans le préambule les champs `\title{...}`, `\author{...}`, `\date{...}` et de placer dans le texte `\maketitle`.

4.2 Texte

La **taille du texte** peut être modifiée via les commandes suivantes, ce qui donne respectivement : a a a a a a a a a a avec les commandes :

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

Ces commandes restent actives tant qu'une autre ne l'annule pas. Une solution est de placer la commande suivie du texte concerné entre accolades.

Quant aux **styles de texte**, on distingue du mode `normal` les suivants :

- la fonte *italique* : `\emph`
- le style *penché* : `\textsl`
- la fonte PETITES CAPITALES : `\textsc`
- la fonte machine à écrire : `\texttt`
- la fonte sans sérif : `\textsf`
- le mode gras : `\textbf`

Par ailleurs, même si L^AT_EX gère automatiquement les espaces au sein du document, qu'ils soient horizontaux ou verticaux, il existe des commandes pour forcer la mise en page. De manière générale, pour les **espaces horizontaux**, on peut utiliser la commande `\hskip xx cm`, ainsi que celles du tableau suivant dans les environnements mathématiques.

<code>\!</code>	<code>\emptyset</code>	<code>\,</code>	<code>\:</code>	<code>\;</code>	<code>\quad</code>	<code>\qquad</code>

TABLE 1 – Espaces mathématiques

La commande `\hfill` repousse le texte et comble par du vide.

`\dotfill` comble avec des points et `\hrulefill` avec un trait.

Pour les **espaces verticaux**, on a de manière similaire `\vskip xx cm`, ainsi que les commandes prédéfinies `\smallskip`, `\medskip` et `\bigskip`.

Quant à la mise en page générale, L^AT_EX gère les **sauts de ligne** et les **sauts de page** comme tout le reste. Pour forcer la main, la commande `\` permet d'aller à la ligne et `\newpage` de commencer une nouvelle page. Mais normalement, c'est à éviter.

4.3 Longueurs

D'autre part, il existe un certain nombre de variables prédéfinies dans L^AT_EX de longueur (marges, alinéas, espaces entre paragraphes, ...). Pour donner une nouvelle valeur à cette variable, la commande est la suivante : `\setlength{\var}{val}`. Par exemple, la variable qui gère la taille de l'alinéa est `\parindent`. Pour annuler en début de paragraphe l'alinéa, on utilise `\noindent`; pour en changer la taille, on place en début de document `\setlength{\parindent}{1cm}`. La variable pour l'espace entre paragraphes est `\parskip`. On peut définir une nouvelle longueur par `\newlength{\var}`.

4.4 Compteurs

Toute numérotation (chapitres, figures, ...) cache un compteur défini par L^AT_EX ou par l'utilisateur. Ce compteur est lié à une commande et incrémentée automatiquement à chaque appel de la commande associée.

- La commande `\newcounter{nom}[ref]` permet de définir le compteur.
- `\stepcounter{nom}` incrémente automatiquement d'une unité le compteur.
- `\setcounter{nom}{nb}` initialise le compteur à `nb`.
- `\addtocounter{nom}{nb}` ajoute `nb` au compteur.

Pour faire apparaître la valeur d'un compteur, on ajoute `\the` devant le nom du compteur.

4.5 Tableaux

Pour créer un tableau, on utilise l'environnement `tabular`, qui prend comme paramètre obligatoire la disposition de chaque colonne, parmi les choix suivants :

- `|` : trace un trait vertical entre 2 colonnes ;
- `c` : centre horizontalement le texte dans la colonne ;
- `l` et `r` : aligne à gauche et à droite ;
- `*{nb}{disp}` : duplique `nb` fois la commande `disp` ;
- `p{xx cm}` : impose la largeur de la colonne (aligné à gauche par défaut).

A l'intérieur de l'environnement, le tableau se remplit ligne par ligne, en séparant les colonnes par des `&`. La ligne se termine par `\\`. Pour tracer une ligne horizontale entre deux lignes, on utilise `\hline` (sur toute la longueur) ou `\cline{m-n}` (entre les colonnes `m` et `n`).

Pour centrer sur plusieurs colonnes, la commande `\multicolumn{nb}{disp}` existe. De la même manière, la commande `\multirow{nb}{xx cm}[param]{texte}` à charger grâce au package `multirow` permet de centrer un texte sur `nb` lignes sur une largeur de `xx cm` avec un paramètre d'ajustement vertical.

Enfin, il est possible grâce au package `colortbl` de mettre de la couleur en fond d'une colonne, d'une ligne ou d'une cellule.

```
\begin{center}
\renewcommand{\arraystretch}{1.25}
\begin{tabular}{ccp{12cm}}
\hline
\rowcolor[gray]{0.8}
Semaine 10 & 10. 2 & \textbf{\textsc{Note}} : \ldots \quad \hfill Y. Penel \\
\hline
\rowcolor[gray]{0.9}
Groupe 5 & & \hfill \text{ 'Elève : \dotfill } \\
\hline\hline
\multirow{3}{2cm}[-.8cm]{\fcolorbox[gray]{0.8}{0.9}{Exercice}}
& \multicolumn{2}{l}{Soit  $f$  la fonction définie par}
\\
0 & \text{si } x \notin ]-1,1[ , & \\
\exp\frac{-x^2}{1-x^2} & \text{sinon.} & \\
\end{cases}
\\
& 1. & \text{Montrer que } f \text{ est continue sur } \mathbb{R} . \\
\end{tabular}
```

```

\cline{2-3}
& 2. & Démontrer que  $f$  est de classe  $\mathscr{C}^{\infty}$  sur  $\mathbb{R}$ .
On utilisera pour cela une suite de
pour exprimer les dérivées successives de  $f$ .
\hline
\end{tabular}
\end{center}

```

Semaine 10	10. 2	NOTE : ...	Y. Penel
Groupe 5		Élève :	
Soit f la fonction définie par $f(x) = \begin{cases} 0 & \text{si } x \notin]-1, 1[, \\ \exp \frac{-x^2}{1-x^2} & \text{sinon.} \end{cases}$			
Exercice	1. Montrer que f est continue sur \mathbb{R} .		
	2. Démontrer que f est de classe \mathscr{C}^{∞} sur \mathbb{R} . On utilisera pour cela une suite de pour exprimer les dérivées successives de f .		

4.6 Listes

Les listes sont des environnements qui permettent de séparer en plusieurs points. Elles peuvent être de trois types :

- `itemize` : listes avec des tirets par défaut
- `enumerate` : listes numérotées
- `description` : listes sans symboles

Les listes peuvent bien sûr être imbriquées les unes dans les autres (éviter au-delà de 3). Pour changer le symbole de `itemize`, utiliser la commande `\renewcommand{\labelitemi}{\$bullet\$}`.

Plusieurs paramètres sont modulables par l'utilisateur, comme l'espace entre le symbole et le texte (`\itemindent`), ou entre deux symboles (`\itemsep`), ...

4.7 Et les maths !

La force de L^AT_EX est de rendre les mathématiques gracieuses au sein de ses documents. On liste ici (de manière **non exhaustive**) les commandes les plus utiles pour les maths.

4.7.1 Environnements

Il existe deux types d'environnements pour les maths. Le premier, dont le contenu est placé entre `$. . . $`, est inséré dans la ligne de texte où intervient la commande. Le second, placé entre `\[. . . \]`, permet de centrer le contenu. Certaines commandes ont des rendus différents selon l'environnement choisi, comme les symboles \int (`\int`), \sum (`\sum`), \prod (`\prod`), qui en mode centré donnent \int , \sum et \prod . On peut toutefois forcer la main à L^AT_EX pour avoir un rendu maximal grâce à la commande `\displaystyle`.

4.7.2 Délimiteurs

Pour ce qui est des délimiteurs, à savoir ([{ |), on peut utiliser les commandes `\left` et `\right` (L^AT_EX adapte alors la taille à celle du texte à l'intérieur des délimiteurs) ou, si l'on veut forcer la taille, `\big`, `\Big`, `\bigg` et `\Bigg` :

\emptyset	<code>\left</code>	<code>\big</code>	<code>\Big</code>	<code>\bigg</code>	<code>\Bigg</code>
$\left(\frac{1}{x} + 1\right)$	$\left(\frac{1}{x} + 1\right)$	$\left(\frac{1}{x} + 1\right)$	$\left(\frac{1}{x} + 1\right)$	$\left(\frac{1}{x} + 1\right)$	$\left(\frac{1}{x} + 1\right)$

TABLE 2 – Taille des délimiteurs

4.7.3 Lettres grecques

Toutes les lettres grecques¹ minuscules existent en environnement mathématique (on tape le nom de la variable précédée du `\`) :

$\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega.$

Certaines ont une variante (`\var` devant le nom de la variable) : $\varepsilon, \vartheta, \varrho, \varsigma, \varphi$, et certaines ont leur majuscule (première lettre en majuscule) : $\Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Upsilon, \Phi, \Psi, \Omega$.

4.7.4 Fontes mathématiques

Quelques fontes : *ABCDEFGHIJKLMNOPQRSTUVWXYZ, abc...* par défaut.

- `\mathcal` (uniquement en majuscule) : *ABCDEFGHIJKLMNOPQRSTUVWXYZ*
- `\mathscr` (package `mathrsfs`) : *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z*
- `\mathsf` : *ABCDEFGHIJKLMNOPQRSTUVWXYZ, abc...*
- `\mathrm` : *ABCDEFGHIJKLMNOPQRSTUVWXYZ, abc...*
- `\mathfrak` : *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z, abc...*
- `\mathbb` (uniquement en majuscule) : *ABCDEFGHIJKLMNOPQRSTUVWXYZ*
- `\mathbf` : *ABCDEFGHIJKLMNOPQRSTUVWXYZ, abc...*
- `\boldsymbol` : *ABCDEFGHIJKLMNOPQRSTUVWXYZ, abc..., \alpha\beta\gamma...*

4.7.5 Commandes

Quelques opérateurs mathématiques à connaître : TABLES 3 et 4.

4.7.6 Matrices

L'environnement `array` permet de construire des tableaux en mode mathématique. Sa structure est similaire à `tabular`. Les arguments obligatoires sont `c`, `l` et `r`, utilisables autant de fois qu'il y a de colonnes.

1. L'alphabet grec est à connaître par cœur en maths!

Racine	Fraction	Puissance	Indice	Coeff. binom.
<code>\sqrt[n]{x}</code>	<code>\frac{num}{denom}</code>	<code>{(x^2+1)}^{n+1}</code>	<code>u_{n+1}</code>	<code>\binom{n}{p}</code>
$\sqrt[n]{x}$	$\frac{num}{denom}, \frac{num}{denom}$	$(x^2 + 1)^{n+1}$	u_{n+1}	$\binom{n}{p}, \binom{n}{p}$
Max.	Sup.	Limite	Logarithme	Cosinus
<code>\max_n</code>	<code>\sup_n</code>	<code>\lim_{x\to 0}</code>	<code>\ln</code>	<code>\cos</code>
\max_n, \max_n	\sup_n, \sup_n	$\lim_{x \rightarrow 0}, \lim_{x \rightarrow 0}$	\ln	\cos
Apparence	Inclusion	Union	Intersection	Inégalité
<code>\in, \notin</code>	<code>\subset, \subseteq</code>	<code>\cup, \bigcup</code>	<code>\cap</code>	<code>\leqslant, \geqslant</code>
\in, \notin	\subset, \subseteq	\cup, \bigcup	\cap	\leq, \geq
Pour tout	Il existe	Différent	Approx.	Equivalence
<code>\forall</code>	<code>\exists</code>	<code>\neq</code>	<code>\approx</code>	<code>\sim</code>
\forall	\exists	\neq	\approx	\sim

TABLE 3 – Quelques commandes usuelles en maths

<code>\rightarrow</code>	<code>\to, \rightarrow</code>	<code>\longrightarrow</code>	<code>\longrightarrow</code>
<code>\Rightarrow</code>	<code>\Rightarrow</code>	<code>\Longrightarrow</code>	<code>\Longrightarrow</code>
<code>\Leftrightarrow</code>	<code>\leftrightharrow</code>	<code>\Leftrightarrow</code>	<code>\Longleftarrow</code>
<code>\mapsto</code>	<code>\mapsto</code>	<code>\longmapsto</code>	
<code>\hookrightarrow</code>	<code>\xrightarrow[n \rightarrow +\infty]{\mathbb{R}}</code>	<code>\xrightarrow[n \rightarrow +\infty]{\mathbb{R}}</code>	<code>\xrightarrow[n \rightarrow +\infty]{\mathbb{R}}</code>

TABLE 4 – Flèches

Le package `amsmath` fournit des environnements plus élaborés que le précédent, avec des délimiteurs intégrés : `pmatrix` pour des parenthèses, `bmatrix` pour des crochets et `vmatrix` pour des barres verticales.

4.7.7 Equations et références

Outre la version centrée simple du mode mathématique, il est possible de numéroter ses **équations** avec une structure plus ou moins complexe :

- pour des équations simples, l'environnement `equation` est parfaitement adapté ; la numérotation peut être indépendante ou liée aux numéros de chapitres, de sections, ...
- pour une suite d'égalités alignées, on utilise l'environnement `align` ; on utilise le symbole `&` pour aligner les termes situés à sa droite et `\\` pour terminer la ligne. Chaque ligne est numérotée. Pour éviter la numérotation d'une ligne, on place la commande `\nonumber` (ou `\notag` avec `amsmath`). La version étoilée élimine toute numérotation.
- lorsqu'une égalité doit se prolonger, on dispose de l'environnement `multline` (on passe à la ligne suivante grâce à `\\`). Il n'y a alors qu'une numérotation pour le bloc.
- les environnements précédents ont une numérotation numérique. Pour avoir des numérotations partielles, on utilise l'environnement `subequations`. Ce n'est pas un environnement mathématique : il faut donc parmi les précédents pour placer du texte mathématique. Toutes les équations, placées dans le même environnement ou non, ont des numérotations du type (1.a), (1.b), ...
- le package `cases` fournit l'environnement `subnumcases` qui englobe les équations dans une accolade, et numérote partiellement chaque équation.

La numérotation peut être linéaire tout au long du document, mais il est possible de réinitialiser la numérotation au début de chaque chapitre ou chaque section en utilisant la commande (`amsmath`) `\numberwithin{equation}{chapter}` (le numéro de la partie concernée apparaît aussi).

On peut par ailleurs donner ponctuellement la forme de la numérotation grâce à la commande `\tag{nom}` qui, placée à la fin d'une équation, fait apparaître `nom` à la place du numéro.

Un outil utile et géré par L^AT_EX est la notion de **références**. On peut en effet faire référence au numéro d'un chapitre, d'une section, d'une équation, d'une figure, d'un tableau, d'une note de bas de page grâce au couple `\label{nom}-\ref{nom}`. Placée après une commande numérotée, la commande `\label{nom}` crée une référence à laquelle il peut être fait appel grâce à la commande `\ref{nom}` qui renvoie le numéro concerné. La commande `\pageref{nom}` renvoie la page où est située la référence concernée. Pour les équations, le package `amsmath` fournit la commande `\eqref{nom}` grâce à laquelle les références apparaissent entre parenthèses.

4.7.8 Divers

Deux exemples qui parlent d'eux-mêmes :

$$\underbrace{(f(x) + g'(x)f(x)) \exp g(x)}_{=(f(x)e^{g(x)})'} \leq h(x) \qquad \prod_{\substack{0 \leq i \leq n \\ i \neq k}} \frac{X - a_i}{a_k - a_i}$$

produits par :

```
\underbrace{\bigl(f(x)+g'(x)f(x)\bigr)\exp g(x)}_{=(f(x)e^{g(x)})'}
\leqslant h(x)
\qqquad\qqquad
\prod_{\substack{0 \leqslant i \leqslant n \\ i \neq k}} \frac{X-a_i}{a_k-a_i}
```

4.8 Quelques commandes supplémentaires de mise en page

Voici en vrac des commandes utiles :

- `\rule{larg}{long}` : trace un trait avec la largeur et la longueur souhaitée ;
- le texte est justifié par défaut. Mais ponctuellement, on peut centrer, aligner à gauche ou à droite grâce respectivement aux environnements `center`, `flushleft` et `flushright` ;
- `\footnote{texte}` : fait apparaître le `texte` en note de bas de page ;
- de même, `\marginpar[gauche]{droite}` permet de faire des notes dans les marges ;
- `\verb$texte$` fait apparaître le texte brut (non interprété) ; `$` est un symbole au choix mais qui doit être le même en début et en fin de texte ;
- `\fbox{...}` pour entourer du texte (au plus une ligne) ; en mode mathématique, la commande analogue est `\boxed{...}` ;
- `$... \text{texte} ... $` pour mettre du texte à l'intérieur d'un environnement mathématique ;

4.9 Figures

Il existe en L^AT_EX deux environnements dits flottants, dans la mesure où c'est le compilateur qui décide du meilleur endroit où il doit apparaître, (presque) indépendamment d'où son appel est placé dans le code. Il s'agit des structures **figure** et **table**. Les arguments optionnels pour les deux sont :

- **b** : le flottant est placé en bas de page
- **t** : le flottant est placé en haut de page
- **p** : le flottant est placé sur une page réservée aux flottants
- **h** : le flottant est placé dans la mesure du possible à l'endroit correspondant dans le code

L'option **h** est cependant fragile et il existe l'option **H** du package **float** qui marche à tous les coups. On peut placer une **légende** dans les environnements flottants grâce à la commande `\caption{texte}`.

Pour **inclure des images** (.jpg, .gif, ...), on dispose de `\includegraphics[dim]{nom}`. Pour la dimension, on précise soit la largeur (`width=xx cm`), soit la hauteur (`height=xx cm`). Le package **graphicx** est fortement recommandé.

5 Liens et littérature

Les sites à mettre en marque-page :

- <http://www.ctan.org/> : il y a tout ce que vous pouvez trouver de relatif à L^AT_EX, à commencer par les codes sources des packages, des FAQ, ...
- <http://texcatalogue.sarovar.org/> : catalogue de packages à télécharger, relativement bien mis à jour
- <http://www.texample.net/> : la puissance de TikZ en quelques jolies images

Les ouvrages de référence :

- *The T_EX book*, D.E. KNUTH, Addison-Wesley, 1988.
- *The L^AT_EX companion*, M. GOOSSENS, F. MITTELBACH & A. SAMARIN, Addison-Wesley, 1994.
- *Tout ce que vous avez toujours voulu savoir sur L^AT_EX sans jamais oser le demander*, V. LOZANO, Framabook, 2008.

Deuxième partie

Supplément : BEAMER

Pour réaliser des présentations scientifiques orales, la classe BEAMER est l'outil parfait. Pour apprendre à l'utiliser, il faut passer par la case `beameruserguide.pdf`, c'est-à-dire environ 230 pages. Sinon, il suffit de lire les lignes qui suivent. Mais dans tous les cas, il est vivement conseillé de lire le **paragraphe 5** du `beameruserguide`, car il contient les ingrédients pour faire une bonne présentation !

1 Installation

Si la classe n'a pas été installée lors de l'installation de M^IK^TE^X, il suffit de la télécharger (voir les liens ci-dessus) et de l'installer selon la méthode décrite dans le paragraphe 3.2.

Il faut également installer PGF à partir de l'adresse http://www.texample.net/tikz/builds/download_latest_build ainsi que le package `xcolor`.

2 Préambule

La classe BEAMER charge automatiquement certains packages comme `xcolor` ou `hyperref`. Si ces packages requièrent des options, il suffit de les préciser dans la déclaration. Par exemple :

```
\documentclass[xcolor={table,dvipsnames}]{beamer}
```

permet d'utiliser `colortbl` pour les tableaux, et les couleurs de type `dvipsnames`.² Les packages usuels peuvent être chargés ensuite.

Comme de manière générale dans L^AT_EX, on peut soit laisser le compilateur gérer la mise en page, soit créer ses propres éléments de présentation. On ne donne ici que les éléments de base.

La classe BEAMER demande de spécifier un thème, une couleur de thème, un thème de fond, . . . , que l'on impose comme suit (dans le préambule) :

```
\usetheme{AnnArbor}
\usecolortheme{dolphin}
\useoutertheme{split}
```

Se reporter aux pages 136-143 du `beameruserguide` pour choisir son thème.

Les commandes habituelles pour entrer les données de la présentation restent les mêmes, mais avec une commande supplémentaire : `\author[alias]{nom}`, `\date{date}`, `\title[alias]{nom}`, `\institute[alias]{nom}`, `\titlegraphic{}` où l'alias est le nom qui apparaît dans les en-têtes et/ou pieds de page, alors que le nom apparaît en entier sur la première page.

2. cf. documentation `xcolor` pour un aperçu des couleurs proposées.

3 Quelques commandes utiles pour le rendu

En premier lieu, pour supprimer les symboles de navigation placés par défaut en bas à droite : `\setbeamertemplate{navigation symbols}{}`.

Pour justifier le texte des diapos, utiliser le package `ragged2e` et la commande `\justifying`.

Pour structurer la présentation, on peut faire appel aux commandes `\section{}` et `\subsection{}` à l'extérieur des `frames`.

Pour mettre en valeur un mot, les commandes `\alert{texte}` et `\structure{texte}` rendent le texte en couleur.

4 Diapos

Pour créer une diapositive (`frame`), on utilise l'environnement du même nom :

```
\begin{frame}[options]{titre}
Texte
\end{frame}
```

L'option `plain` permet d'afficher la diapo sur la page entière, supprimant ainsi ponctuellement entêtes et pieds de page. En particulier, on l'utilise pour la page de titre, le texte étant alors `\titlepage`. On peut également faire une diapo avec la table des matières (la commande reste la même).

A l'intérieur des diapos, on peut utiliser les commandes mathématiques, de listes, d'environnements usuels. L'environnement `block` permet de créer une boîte avec un fond coloré.

Pour créer des effets de transparence, il existe différentes possibilités. Dans une même diapo (`frame`), il peut y avoir plusieurs transparents (`slides`) pour créer des apparitions/disparitions de texte :

- la commande `\pause` : le texte placé dessous n'apparaît que lorsque l'on appuie sur une touche ;
- la commande `\only<nb>{texte}` ne montre le texte que dans les transparents `nb`.³ Sur les autres transparents de la diapo, il n'y a même pas d'espace alloué à ce texte ;
- la commande `\visible<nb>{texte}` qui rend le texte visible que sur les transparents spécifiés. Le reste du temps, il y a un blanc de la taille du texte ;
- la commande `\uncover<nb>{texte}` qui a le même rendu que la commande précédente, sauf s'il a été spécifié un effet de transparence via `\setbeamercovered{transparent=20}`. Dans ce cas, lorsque le texte n'est pas affiché, il apparaît en gris clair (selon la transparence demandée).

La majorité des commandes usuelles possèdent alors un argument optionnel pour spécifier les apparitions du texte. Par exemple, dans les listes, `\item<nb>` ne fera apparaître (comme `\uncover`) le texte que sur les transparents spécifiés. La commande `\textbf<nb>{texte}` rendra le texte gras sur le transparent `nb` et normal le reste du temps. Le principe est le même pour `\color<nb>{red}`.

3. `nb` peut être un numéro (2), ou un ensemble (1 – 3).