

Introduction à L^AT_EX

1 Introduction à L^AT_EX

1.1 Historique

L^AT_EX est un *système de composition de documents* et plus exactement un ensemble de macro-commandes destinées à faciliter l'utilisation de T_EX. T_EX (« tèk ») est un puissant programme de mise en forme de documents, basé sur le balisage. Ce programme a été créé par Donald E. Knuth en 1977 pour l'édition scientifique, mathématique en particulier, et reste largement utilisé dans le domaine de l'édition scientifique. T_EX, qui est un langage de description formel, présente des particularités qui le distinguent des autres langages informatiques, dont le fait que son dernier bug a été corrigé... en 1993. De fait, T_EX est un des rares (le seul ?) programme terminé, ce qui est assez rare pour être souligné.

L^AT_EX (« latèk ») est un moteur créé par Leslie Lamport en 1986, reposant sur T_EX, plus facile d'utilisation que T_EX. La version actuelle, datant de 1994 est L^AT_EX 2_ε. Contrairement à T_EX, ce moteur continue lui d'évoluer et la future version devrait ainsi être la version 3.

1.2 T_EX : les qualités

- T_EX est terminé depuis 1993 (plus de bug)
- Il n'y a donc pas de problèmes de compatibilité de versions...
- T_EX appartient au domaine public (stabilité, gratuité, disponibilité)
- T_EX est un langage de description de documents et de programmation
- L'unité de longueur de base est très petite : 0,00039 mm !
- Les documents T_EX sont de simples fichiers texte, ce qui est très pratique pour échanger des documents ou les stocker
- C'est un langage bien adapté à la gestion de longs documents et au traitement des mathématiques
- Ce programme est utilisable en ligne de commande, ce qui permet de l'interfacier facilement...
- Il est possible de définir des macro-commandes (cas typique de L^AT_EX d'ailleurs), ce qui permet de l'étendre facilement.

1.3 T_EX : les défauts

- T_EX est un vieux programme, ce qui peut parfois freiner ses évolutions, notamment vis-à-vis des nouveaux formats de données (fontes, images).
- T_EX nécessite des fontes spéciales.
- T_EX n'est pas adapté à la réalisation de documents type PAO, mais cela est le cas de la majeure partie des logiciels les plus similaires à L^AT_EX, les traitements de texte
- T_EX n'est pas interactif, ce qui peut dérouter au premier abord...

- La programmation sous $\text{T}_{\text{E}}\text{X}$ n'est pas toujours aisée et demande une période d'apprentissage

1.4 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: les apports

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est une surcouche de $\text{T}_{\text{E}}\text{X}$ incluant un moteur et des macros ainsi que certaines notions assez pratiques, comme la notion d'environnements, assez proche des éléments SGML (et donc de HTML par exemple). Par rapport à $\text{T}_{\text{E}}\text{X}$, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ permet également le support d'arguments optionnels dans les macros, la notion de modules (*packages*) qui permettent de l'étendre. Du coup, la communauté d'utilisateurs a créé de nombreux packages, notamment en :

- sciences : chimie, physique, mathématiques, ...
- loisirs : musique,
- langues : arabe, hébreu, langues asiatiques

1.5 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: marche à suivre

Pour créer un document en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$:

1. Installer le paquet `texlive`, qui contient une *distribution* $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.
2. Installer aussi le paquet `texlive-lang-french` qui contient le support du français, ou le paquet `texlive-full` qui a des dépendances sur tous les paquets TeXLive.
3. saisir le texte source et le sauvegarder dans un fichier à l'extension `.tex`
4. compiler le document à l'aide de `pdflatex` (`pdflatex test.tex`), qui génère un document PDF.
5. visualiser le document PDF produit

Il est également possible de générer des documents DVI avec la commande `latex`, puis de les convertir en PDF (`dvipdf`) ou en PostScript (`dvips`).

2 Écriture de documents $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Lors de cette séance, il vous est simplement demandé d'essayer les différentes commandes mentionnées dans le sujet, et donc d'écrire votre premier document $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

2.1 Introduction

Quelques généralités à propos de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$:

- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est sensible à la casse
- Quelques caractères ont des significations particulières sous $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: `$`, `&`, `%`, `#`, `{`, `}`, `_`, `\`, `^`, `~`
 - `%` : le reste de la ligne est un commentaire
 - `\` : introduit une commande (un identificateur sans chiffre)
 - `~` : espace insécable
- Pour pouvoir utiliser ces caractères dans un document, il faut les préfixer d'un antislash (`\`), sauf pour `\` qu'il faut écrire `\textbackslash`

2.2 Squelette minimum

Le squelette minimum d'un document L^AT_EX est :

```
\documentclass{article}
% En-tête du document
\begin{document}
Corps du document
\end{document}
```

- La commande `\documentclass` permet de choisir un modèle de document : `book`, `report`, `article`, `letter`
- Le corps du document se situe dans l'environnement `document`, délimité par `\begin{document}` ... `\end{document}`
- L'en-tête du document permet de déclarer des extensions (*packages*), notamment celles relatives au français
 - encodage de caractères
 - `\usepackage[utf8]{inputenc}`
 - `\usepackage[T1]{fontenc}`
 - support du français
 - `\usepackage[french]{babel}` (recommandé) ou
 - `\usepackage{french}`

2.3 Généralités

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[OT1]{fontenc}

\usepackage[french]{babel}

\begin{document}

Hello !
Ceci est une seconde phrase...
Et une      troisième
phrase...
Ici, du texte permettant de meubler un peu ma ligne et voir plus
facilement le changement de paragraphe.

Ici, du texte permettant de meubler un peu ma ligne et voir plus
facilement le changement de paragraphe.
Et maintenant le caractère \%, à ne pas % ici, un commentaire
confondre avec la marque de commentaire qui n'est pas préfixée du
\textbackslash.

Nouveau paragraphe... et donc indentation !
\end{document}
```

Quelques remarques :

- Les espaces multiples sont ignorées
- Les retours à la ligne dans le source sont considérés comme des espaces dans le résultat
- Une ligne vide dans le source est considérée comme un changement de paragraphe
- Avec le support du français, les espaces du source sont automatiquement converties en différentes sortes d'espaces dans le résultat (insécable, $\frac{3}{4}$...)
- `\\` permet de passer à la ligne (sans changer de paragraphe)

2.4 Les caractères

En dehors des caractères disponibles sur les claviers, il est possible de saisir des séquences décrivant des caractères (le plus simple reste bien sûr de saisir un caractère au clavier s'il s'y trouve) :

Séq.	Donne	Séq.	Donne
<code>\`{a}</code>	à	<code>\^{a}</code>	â
<code>\"a</code>	ä	<code>\,{C}</code>	Ç
<code>\v{v}</code>	ř	<code>\"i</code>	ï
<code>\u{o}</code>	ö	<code>\d{o}</code>	o
<code>\oe</code>	œ	<code>\AA</code>	Å
<code>\O</code>	Ø	<code>\pounds</code>	£
<code>---</code>	—	<code><<</code>	«

2.5 Tailles de caractères

En \LaTeX , les tailles de caractères ne se définissent pas dans l'absolu, mais de façon relative. Voilà des exemples de changement de taille (les accolades `{}` et `}` permettent de définir un bloc confinant les modifications de taille) :

- `\tiny` : De plus en plus grand
- `\scriptsize` : De plus en plus grand
- `\footnotesize` : De plus en plus grand
- `\small` : De plus en plus grand
- `\normalsize` : De plus en plus grand
- `\large` : De plus en plus grand
- `\Large` : De plus en plus grand
- `\LARGE` : De plus en plus grand
- `\huge` : De plus en plus grand
- `\Huge` : De plus en plus grand

2.6 Styles de caractères

Les changements de styles se définissent d'une manière assez similaire :

- **gras** : commande `\textbf{...}`
- *italique* : commande `\textit{...}`
- PETITES CAPITALES : commande `\textsc{...}`
- machine à écrire : commande `\texttt{...}`
- Pour changer de police de caractères : plus compliqué
- Mais si le changement est pour tout le document, cela reste simple. On écrira `\usepackage{times}` dans l'en-tête du document pour tout passer en « times » par exemple

2.7 Environnements de formatage

Un environnement `bidule` est défini en \LaTeX par `\begin{bidule} ... \end{bidule}`. Il existe des environnements d'alignements :

- `center` : centre le texte contenu
- `flushleft` : aligne à gauche
- `flushright` : aligne à droite

Et les environnements relatifs aux listes (chaque item est introduit grâce à `\item`) et les listes peuvent naturellement s'imbriquer :

- `itemize` : liste non ordonnée
- `enumerate` : liste ordonnée

2.8 Gestion des titres

L^AT_EX facilite grandement la gestion des titres, dont il suffit de préciser le niveau hiérarchique : (`\part{...}`, `\chapter{...}`, `\section{...}`, `\subsection{...}`, `\subsubsection{...}`, `\paragraph{...}`, `\subparagraph{...}`) et automatiquement :

- les espaces avant et après les titres sont gérés
- la mise en forme du titre est effectuée
- la numérotation est faite (sauf si les commandes sont appelées avec le suffixe `*`, comme `\chapter*{...}` par exemple)

2.9 Quelques autres commandes

- Si les titres sont utilisés, il est possible d'insérer une table des matières, en insérant à l'endroit souhaité `\tableofcontents`
- Vous pouvez aussi définir un titre, un auteur, et une date, et générer automatiquement un titre. Voir https://en.wikibooks.org/wiki/LaTeX/Title_Creation
- La commande `\newpage` permet de sauter une page
- La commande `\footnote{...}` permet de définir une note de bas de page ¹
- La commande `\today` donne la date courante
- Les commandes `\label{...}` et `\ref{...}` permettent d'associer des étiquettes à des éléments et d'y faire référence ensuite

2.10 Les mathématiques

C'est la grande spécialité de L^AT_EX ! Deux environnements permettent de passer en mode mathématique : `$...$` : pour de petites formules et `$$...$$` : pour des formules affichées seules sur une ligne. Dans ce mode, L^AT_EX change de style de mise en forme (espacements, police, ...) et de nouveaux symboles disponibles : $\alpha, \beta, \Phi, \Sigma, f, \overline{phil}, \tilde{o}, \nabla, \neq, \in, \perp$, etc... Ainsi :

1. Comme ça...

Il est possible de placer des équations directement dans le texte, de cette manière : $\alpha = \frac{\sqrt{\pi}}{\sqrt{\lambda}}$. Cela n'a pas de conséquence sur la mise en forme des lignes de ce paragraphe...

Il est également possible de placer des équations en évidence et on peut même demander une numérotation automatique et faire référence ensuite à des formules de cette manière : équation `\ref{eq}`.

```
\begin{equation}
\label{eq}
\underbrace{
a_{i,j}^{\frac{1}{2}} = \sqrt{\frac{1}{n} \times \int_{n=a}^c \Delta f_n h_{i,j,n}} = \sqrt{\frac{2^{\frac{i}{2}}}{2^p} \Big(\sum_{n=a}^b \Delta f_n - \sum_{n=b}^c \Delta f_n\Big)}
}_ {\mathrm{dont\ la\ racine\ est\ triviale}}
\end{equation}
```

Donnera :

Il est possible de placer des équations directement dans le texte, de cette manière : $\alpha = \frac{\sqrt{\pi}}{\sqrt{\lambda}}$. Cela n'a pas de conséquence sur la mise en forme des lignes de ce paragraphe... Il est également possible de placer des équations en évidence et on peut même demander une numérotation automatique et faire référence ensuite à des formules de cette manière : équation 1.

$$a_{i,j}^{\frac{1}{2}} = \underbrace{\sqrt{\frac{1}{n} \times \int_{n=a}^c \Delta f_n h_{i,j,n}} = \sqrt{\frac{2^{\frac{i}{2}}}{2^p} \left(\sum_{n=a}^b \Delta f_n - \sum_{n=b}^c \Delta f_n \right)}}_{\text{dont la racine est triviale}} \quad (1)$$

2.11 Les images

- Il est possible d'inclure des images et de les manipuler grâce au package `graphicx`
- Une commande pour insérer les images `\includegraphics[options]{nom}`
- Les options permettent de fixer l'angle, la largeur, la hauteur, l'échelle, la sélection d'une partie de l'image à insérer
- Si vous générez des documents PDF avec `pdflatex`, les formats d'image courant sont acceptés (JPG, PNG, ...)
- \LaTeX propose aussi un environnement `figure` permettant
 - d'insérer une ou plusieurs images **flottantes**
 - d'associer une légende aux images
 - de donner un label
- Exemple

```

\begin{figure}[htbp]
  \includegraphics[width=.9\linewidth]{yo.png}
  \caption{Ma légende.}
  \label{fig:yo}
\end{figure}

```

2.12 Les tableaux

- Plusieurs environnements sont disponibles, dont tabular
- Exemple

```

\begin{tabular}{|c|lr|}
  \hline
  Yo & Hello & Bonjour\\
  \hline
  Bye & Good-bye & Au revoir\\
  \hline
\end{tabular}

```

Yo	Hello	Bonjour
Bye	Good-bye	Au revoir

- Explications
 - c défini un champ centré, l un champ aligné à gauche, r un champ aligné à droite
 - | trace des lignes séparatrices verticales
 - & est le séparateur de champs
 - \hline trace des lignes séparatrices horizontales

2.13 Dimensions de la page

Le plus simple est d'utiliser le package vmargin :

```

\usepackage{vmargin}
\setpapersize{A4}
\setmargins{15mm}{15mm}{180mm}{260mm}{0pt}{0mm}{0pt}{1.2cm}

```

Signification des paramètres : `\setmargins{leftmargin}{topmargin}{textwidth}{textheight}{headheight}{headsep}{footheight}{footskip}`

2.14 Les macros

- Il est possible de définir de nouvelles commandes \LaTeX grâce à `\newcommand`
- Exemple


```

\newcommand{\yo}[1]{\textbf{\small #1}}
\yo{Hello}

```
- Les commandes définies peuvent éventuellement accepter plusieurs arguments ([1]) qui peuvent ensuite être utilisés dans la macro (#1)
- Sur le même principe, il est possible de définir de nouveaux environnements
- Exemple


```

\newcommand{\yo}[1]{\textbf{\Large #1}}

\begin{tabular}{|l|c|c|}
\hline
\yo{Jour} & \yo{Débit} & \yo{Crédit}\\
\hline
Lundi & 125 000 & 64 000\\
Mardi & 118 000 & 97 000\\
Mercredi & 108 000 & 76 000\\
\hline
\end{tabular}

```

Donnera :

Jour	Débit	Crédit
Lundi	125 000	64 000
Mardi	118 000	97 000
Mercredi	108 000	76 000

3 Et encore...

3.1 Distributions L^AT_EX

- Tout comme Linux, il existe de nombreuses distributions L^AT_EX
 - sous Linux/Unix : te_EX, T_EXlive
 - sous Windows : Mik_EX, Win-GUT
 - sous Macintosh : Mac-GUT, CMac_EX
- Certaines distributions sont plus complètes que d'autres (packages, éditeurs, outils annexes...)

3.2 Et Emacs dans tout ça... ?

Emacs permet d'éditer facilement les sources T_EX et L^AT_EX à l'aide du package AucT_EX, qui fait généralement l'objet d'un package à part sous les distributions Linux. AucT_EX facilite l'accès à L^AT_EX aux débutants grâce à un menu permettant, sous X, de se familiariser avec l'ensemble des fonctions de base. Pour les plus avancés, ce même package fait gagner un temps fou grâce à une multitude de raccourcis relativement intuitifs. Pour activer ce package, il suffit d'ajouter la ligne suivante à son .emacs (ou tout autre fichier de configuration de Emacs) : (require 'tex-site)

Deux raccourcis (clés) principaux sont alors disponibles :

- C-c C-e : qui permet d'insérer des environnements (le mini-buffer propose des choix et il est possible d'utiliser la complétion automatique).
- C-c C-c : qui permet d'activer des commandes externes, permettant par exemple de compiler et de visualiser les fichiers résultats.

Le projet Vim-LaTeX (<http://vim-latex.sourceforge.net>) fournit des outils similaires pour Vim (paquet vim-latexsuite dans Debian et Ubuntu).

3.3 Aller plus loin

Il existe de nombreux livres en ligne sur \LaTeX , comme par exemple « Tout ce que vous avez toujours voulu savoir sur \LaTeX sans jamais oser le demander », un livre libre aux éditions Framabook (<http://www.framabook.org/latex.html>), ou un *Wikibook* (version française : <http://fr.wikibooks.org/wiki/LaTeX>; version anglaise : <http://en.wikibooks.org/wiki/LaTeX>).

Une bonne manière d'apprendre \LaTeX est d'échanger des astuces avec vos enseignants, qui l'utilisent en général pour préparer leurs supports de cours (et leurs *slides*, avec `latex-beamer`).

3.4 Correction orthographique

Pour vérifier l'orthographe d'un document \LaTeX , il est possible d'utiliser `aspell`.

— Pour un document en français :

```
aspell -encoding=utf-8 -t -l fr -c mondocument.tex
```

— Pour un document en anglais :

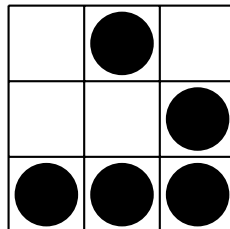
```
aspell -encoding=utf-8 -t -l en -c mondocument.tex
```

3.5 Figures

Pour dessiner des figures, plusieurs solutions peuvent être utilisées :

— Utiliser un éditeur externe comme Inkscape, Xfig ou Dia qui permettra de générer une figure à un format vectoriel (EPS, PDF), à intégrer avec `\includegraphics`;

— Dessiner la figure directement en \LaTeX , par exemple avec PGF & TikZ.



```
\begin{tikzpicture}[thick]
\draw (0,0) grid (3,3);
\foreach \c in {(0,0), (1,0), (2,0), (2,1), (1,2)}
  \fill \c + (0.5,0.5) circle (0.42);
\end{tikzpicture}
```

FIGURE 1 – Exemple de figure simple avec TikZ

3.6 Éditeurs graphiques

Il existe plusieurs éditeurs WYSIWYG (*What You See Is What You Get*) pour \LaTeX : LyX, Kile, TeXmacs, etc.

3.7 Raffraichissement automatique du document généré

Vous pouvez utiliser `latexmk` pour rafraichir automatiquement le document généré à chaque sauvegarde.

Il faut installer le paquet `latexmk`.

Puis le configurer, en créant un fichier de configuration `.latexmkrc` dans votre répertoire personnel (`$HOME`), avec le contenu suivant :

```
$pdf_previewer = 'evince %O %S';  
$pdflatex = "pdflatex --shell-escape %O %S";  
$sleep_time = 1;  
$pdf_mode = 1;
```

Enfin, vous pouvez lancer `latexmk` avec l'option `-pvc` (*preview continuously*) : `latexmk -pvc mondocument`.