

# Cours 1 - Introduction à L<sup>A</sup>T<sub>E</sub>X

L3 Mathématiques 2017/18

15 janvier 2018

# Pourquoi $\text{\LaTeX}$ ?



$\text{\LaTeX}$  est un **langage** de composition de documents, créée en 1983 par Leslie Lamport, successeur de  $\text{\TeX}$  (Donald Knuth, 1977).

Par rapport à un *word processor* plus classique, la première différence est que ce que l'on écrit ne correspond pas à ce que l'on voit, de la même façon que le code *html* n'est pas ce que l'on visualise lorsque on est sur un navigateur. Mais comme pour le *html*, son potentiel est que du texte brut produit des résultats très complexes (sans devoir se casser trop la tête).

$\text{\LaTeX}$  *représente aujourd'hui le langage d'écriture de toute la communauté mathématique* et (en partie) scientifique.



Voici la page précédente, en format de texte brut :

```
\begin{frame}{Pourquoi \LaTeX ?}
  \begin{minipage}{.2\textwidth}
    \includegraphics[width=50px]{lamport.jpg}\[\lem]
    \includegraphics[width=50px]{knuth.jpg}
  \end{minipage}
  \begin{minipage}{.8\textwidth}
    \LaTeX est un \alert{langage} de composition de
    documents, créée en 1983 par Leslie Lamport, successeur de
    \TeX (Donald Knuth, 1977).
```

Par rapport à un `\emph{word processor}` plus classique, la première différence est que ce que l'on écrit ne correspond pas à ce que l'on voit, de la même façon que le code `\emph{html}` n'est pas ce que l'on visualise lorsque on est sur un navigateur. Mais comme pour le `\emph{html}`, son potentiel est que du texte brut produit des résultats très complexes (sans devoir se casser trop la tête).

```
\smallskip
```

`\LaTeX` `\emph{représente aujourd'hui le langage d'écriture de toute la communauté mathématique}` et (en partie) scientifique.

```
\end{minipage}
\end{frame}
```

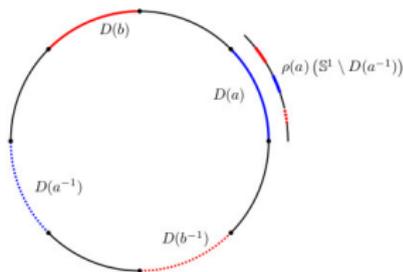


Figure 2.1: Classical ping-pong on two generators

pairwise disjoint open sets  $D(a) \subset \mathbb{S}^1$ ,  $a \in S \cup S^{-1}$ , each of which has finitely many connected components, and such that  $\rho(a)(\mathbb{S}^1 \setminus D(a^{-1})) \subset \overline{D(a)}$ . We further assume that if  $I$  and  $J$  are any connected components of  $D(a)$ , then  $\overline{I} \cap \overline{J} = \emptyset$ .

We call the sets  $D(a)$  the *ping-pong domains* for  $\rho$ .

A similar definition is given in [21], with the additional requirement that ping-pong domains be closed. The above, more general definition is more natural for our purposes, although we will later introduce Convention 3.7 to reconcile the two. The reader may notice that, for a given ping-pong action  $\rho$  of  $(G, S)$ , there can be many choices of sets  $D(a)$  satisfying the property in Definition 2.1. For instance, if  $\rho$  is a ping-pong action such that  $\bigcup_{a \in S \cup S^{-1}} \overline{D(a)} \neq \mathbb{S}^1$ , then one may choose an arbitrary open set  $I$  disjoint from  $\bigcup_{a \in S \cup S^{-1}} \overline{D(a)}$  and replace  $D(a_1)$  with  $D(a_1) \cup I$ , leaving the other domains unchanged. These new domains still satisfy  $\rho(a)(\mathbb{S}^1 \setminus D(a^{-1})) \subset \overline{D(a)}$ . Later we will adopt a convention to avoid this kind of ambiguity.

**Motivation: why ping-pong actions?** The classical ping-pong lemma implies that ping-pong actions are always faithful, and a little more work shows that the action is determined up to semi-conjugacy by a finite amount of combinatorial data coming from the cyclic ordering and the images of the connected components of the sets  $D(a)$  (see Definition 2.4 and Lemma 3.4 below, or [23, Thm. 4.7]). In particular, one can think of ping-pong actions as the family of “simplest possible” faithful actions of  $F_n$  on  $\mathbb{S}^1$ , and it is very easy to produce a diverse array of examples. Perhaps the best-known examples are the actions of discrete, free subgroups of  $\mathrm{PSL}(2, \mathbb{R})$  on  $\mathbb{R}P^1$ . For these actions, one can choose domains  $D(a)$  with a single connected component. Figure 2.1 shows an example of the dynamics of such an action of  $F_2 = \langle a, b \rangle$ .

$\LaTeX$  permet de structurer un texte mathématique :

- Figures
- Maths
- Paragraphes
- Liens
- Références
- et bien plus !

# Installer L<sup>A</sup>T<sub>E</sub>X sur sa machine

**Écriture** : Un fichier L<sup>A</sup>T<sub>E</sub>X est un simple fichier de texte, avec extension `.tex`. Néanmoins, il existe des divers *logiciels* d'écriture en L<sup>A</sup>T<sub>E</sub>X, qui simplifient l'écriture avec des raccourcis clavier et dans les menus. Ce qui est plus, ils permettent aussi de lancer des commandes de *compilation* du fichier, ce qui donne le produit final souhaité, comme un joli fichier `.pdf`.

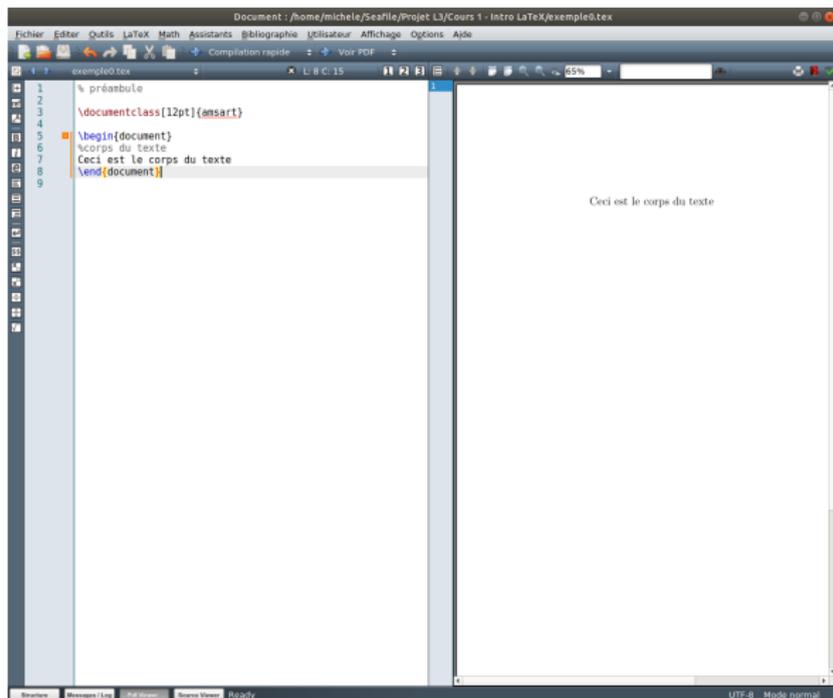
Personnellement j'utilise **Texmaker**, puisqu'il est disponible pour tout système d'exploitation : Linux, OS X, Windows. Dépendant des systèmes d'exploitation, on peut trouver des logiciels plus complets (un vrai geek n'utiliserait qu'**Emacs**). La page *Wikipedia* sur L<sup>A</sup>T<sub>E</sub>X présente une liste assez riche.

On trouve aujourd'hui des nombreux outils en ligne.

**Compilation** : Un fichier `.tex` nécessite de compilation. Aujourd'hui on utilise PDFLaTeX. Cela nécessite d'être présent sur la machine, et son installation dépend des systèmes d'exploitation (ex. : MikTeX sous Windows, texlive sous Linux, MacTeX sous OS X). Les logiciels permettent de lancer la compilation avec des raccourcis clavier.

# Mon premier texte L<sup>A</sup>T<sub>E</sub>X

Un fichier .tex est composé de **deux parties** : le **préambule** et le **corps du texte**. Le texte précédé de % est **commenté**



The screenshot shows a LaTeX editor window with a menu bar (Fichier, Edit, Outils, LaTeX, Math, Assistants, Bibliographie, Utilisateur, Affichage, Options, Aide) and a toolbar. The main window is split into two panes. The left pane shows the source code of a LaTeX document:

```
1 % préambule
2
3 \documentclass[12pt]{ansart}
4
5 \begin{document}
6 %corps du texte
7 Ceci est le corps du texte
8 \end{document}
9
```

The right pane shows the rendered output of the document, which is a white page with the text "Ceci est le corps du texte" centered.

# Compilation

Un fichier `.tex` est compilé par une commande du type

```
pdflatex fichier.tex
```

Si le fichier utilise des liens intertextuels, ou une table des matières, une *deuxième compilation* sera nécessaire pour que le document `.pdf` soit correct.

Si le fichier a une liste de références, il sera aussi nécessaire de compiler sa *bibliographie* (`bibtex`), et répéter la compilation `pdflatex` plusieurs fois.

Les fichiers complémentaires (bibliographie, figures, etc.) doivent être (d'habitude) dans le même dossier que le fichier `.tex`

Les **erreurs** de compilations apparaissent dans le fichier `.log` et ils sont normalement affichés par le logiciel. Il y en a qui ne sont pas grave, à considérer plutôt comme des avertissements (`badboxes`).

# Écrire du texte

Le corps du texte est encadré entre un `\begin{document}` et un `\end{document}`. Pour produire un texte en Français, en particulier avec les caractères spéciaux

é, è, à, ù, ç, É, À, ü, ï,

et les règles de ponctuation, on aura besoin de divers **pacakges** (extensions), dans le préambule :

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

```
\usepackage{lmodern}
```

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

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

(cela peut dépendre du système d'exploitation). De manière universelle, on peut reproduire tout accent avec une commande spécifique : par exemple, é peut s'écrire en utilisant `\'e`

## Styles de texte

La **police** du texte est d'habitude uniforme. Elle est choisie dans le préambule. Par défaut, la police est *Computer Modern*. On peut aussi choisir

<i>Utopia</i>	fourier
<i>Times</i>	mathptmx
<i>mathpazo</i>	Palatino
	iwona

que l'on choisit en écrivant `\usepackage{fourier}` dans le préambule. Chercher plus de détails sur internet.

## Styles de texte

La **police** du texte est d'habitude uniforme. Elle est choisie dans le préambule. Par défaut, la police est *Computer Modern*. On peut aussi choisir

<i>Utopia</i>	fourier
<i>Times</i>	mathptmx
<i>mathpazo</i>	Palatino
	iwona

que l'on choisit en écrivant `\usepackage{fourier}` dans le préambule. Chercher plus de détails sur internet.

Le **style** de texte se modifie localement : on peut écrire en

*italique* `\textit{●}`, **rélie**f `\textbf{●}`, MAJUSCULE `\textsc{●}`,

mais aussi `\texttt{●}`, `\textsf{●}`, `\textsl{●}`.

Cependant, la **classe** et les **réglages** dans le préambule du document gèrent les changements de style dont on a besoin. Pour cela, on privilégiera `\emph{●}` à `\textit{●}`, puisqu'il est capable de s'adapter au style du paragraphe.

Il est important de connaître la manière dont  $\text{\LaTeX}$  gère les espaces. Voici quelques règles générales :

**Retour à la ligne** : un simple retour à la ligne n'a comme seul effet, celui de séparer le dernier mot de la première ligne du premier mot de la deuxième. Écrire deux mots ou

deux  
mots

donne le même texte : *deux mots*. Pour commencer un nouveau paragraphe, on fera un double retour à la ligne (ou même plus, le résultat sera le même). Pour obtenir un espace vertical plus large, on utilisera des commandes comme

`\smallskip, \medskip, \bigskip, \vspace{3cm}`

La commande `\\` commence une nouvelle ligne de texte, sans terminer le paragraphe. On peut choisir la largeur, en écrivant par exemple `\\[2cm]`

**Espaces** : Il peut il y avoir plusieurs espaces entre deux mots,  $\text{\LaTeX}$  n'en produira qu'un seul. Écrire deux mots ou deux mots donne le même texte. Pour gérer les longueurs des espaces, on utilisera des commandes appropriées :

espace insécable	<code>\~</code>
espace fine	<code>\,</code>
espace moyenne	<code>\:</code>
espace large	<code>\;</code>
espace plus large	<code>\quad</code>
espace personnalisé	<code>\hspace{2cm}</code>

Voir plus de détails en ligne.

**Nouvelle page** : `\newpage`

# Positionnement du texte

Le texte est présenté par défaut en mode « justifié ». On peut changer le positionnement du texte en encadrant le texte entre

<code>\begin{flushleft}</code>	<code>\begin{center}</code>	<code>\begin{flushright}</code>
texte	texte	texte
<code>\end{flushleft}</code>	<code>\end{center}</code>	<code>\end{flushright}</code>

# Écrire des maths

On écrit toute formule « mathématique » en l'encadrant par  $(\text{texte})$ .

Les commandes  $\[(\text{texte})\]$ ,  $\begin{equation}(\text{texte})\end{equation}$  disposent la formule centrée dans le texte (le deuxième permet de *numéroter*) :

$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$  produit  $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ .

Mettre la même formule entre  $\[ \]$  donne

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$\$ \$$  accomplit la même fonction que  $\[ \]$  (forme *obsolète*)

Les espaces ne sont pas pris en compte dans l'environnement « math ». Si on veut écrire du texte à l'intérieure d'une formule, on utilisera la commande  $\text{text}\{$

Les logiciels d'écriture présentent des raccourcis souris pour écrire tout type de symboles. On trouvera aussi toute indication nécessaire en ligne (voir par exemple la ressource [detexify](#))

- Tout comme pour le corps du texte normal, on peut modifier le style texte qu'on écrit avec des commandes comme `\mathrm{}`, `\mathsf{}`, `\mathit{}`, `\mathcal{}`, `\mathbf{}`, `\mathbb{}`. Par exemple,  $\mathbb{R}$  s'écrit `\mathbb{R}`.
- On modifie aussi les symboles avec des « accents mathématiques » comme `\hat{}`, `\tilde{}`, `\bar{}`. Par exemple,  $\overline{X}$  s'écrit `\overline{X}`.

*Ces commandes sont amplement utilisées dans un texte mathématique pour diversifier les notations.*

On écrit des matrices avec la commande

```
\begin{matrix}
a & b \\
c & d
\end{matrix}
```

ce qui rend  $\begin{matrix} a & b \\ c & d \end{matrix}$

Si l'on souhaite écrire une matrice entre parenthèses, on utilisera alors la commande

`pmatrix` :  $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Un texte peut aussi être structuré par d'autres commandes comme `align`, `multline`, `eqnarray`, toutes ayant leur intérêt spécifique.

# Structurer son texte

Dans le préambule, on spécifiera les **métadonnées** du document : *titre*, *auteurs*, *date*. Cela se fait par les commandes `\title{titre}`, `\author{}`, `\date{}` (`\today` donne la date de la machine). Le titre sera ensuite écrit dans le corps du texte par la commande `\maketitle`.

# Structurer son texte

Dans le préambule, on spécifiera les **métadonnées** du document : *titre*, *auteurs*, *date*. Cela se fait par les commandes `\title{titre}`, `\author{}`, `\date{}` (`\today` donne la date de la machine). Le titre sera ensuite écrit dans le corps du texte par la commande `\maketitle`.

La structure du texte dépend premièrement de la **classe** du document choisie : les classes `amsart` et `article` sont adaptées à des textes de petite et moyenne taille. Ces sont les classes privilégiées pour des *articles*, *notes de cours*, *feuilles de TD*, etc.

Pour écrire des *livres*, on utilise la classe `book`. On peut aussi écrire des *lettres* (`letter`), des CV (`moderncv`). On peut écrire des *présentations* (`beamer`) et *posters* (`beamerposter`). Voir plus d'informations dans les ressources en ligne.

# Structurer son texte

Dans le préambule, on spécifiera les **métadonnées** du document : *titre*, *auteurs*, *date*. Cela se fait par les commandes `\title{titre}`, `\author{}`, `\date{}` (`\today` donne la date de la machine). Le titre sera ensuite écrit dans le corps du texte par la commande `\maketitle`.

La structure du texte dépend premièrement de la **classe** du document choisie : les classes `amsart` et `article` sont adaptées à des textes de petite et moyenne taille. Ces sont les classes privilégiées pour des *articles*, *notes de cours*, *feuilles de TD*, etc.

Pour écrire des *livres*, on utilise la classe `book`. On peut aussi écrire des *lettres* (`letter`), des CV (`moderncv`). On peut écrire des *présentations* (`beamer`) et *posters* (`beamerposter`). Voir plus d'informations dans les ressources en ligne.

Un texte qui a comme classe `amsart`, sera structuré en **sections** (`section`), **sous-sections** (`subsection`), et ainsi de suite : `subsubsection`, `paragraph`.

Une section commence par la commande `\section{titre de la section}`

La table des matières apparaîtra dans le texte par la commande `\tableofcontents`

# Les packages (extensions ou modules)

$\LaTeX$  ne connaît pas toutes les commandes possibles par défaut. Pour cela, on rajoute des extensions dans le préambule avec la commande `\usepackage[]{}{}`. Parmi les plus nécessaires pour produire un texte de maths, on a `amsrefs`, `amsthm`, `amssymb`, `amsmath`, `enumerate`

# Personnaliser les commandes

Dans le préambule on écrit des raccourcis pour des symboles utilisés fréquemment : par exemple, on préfère écrire `\R` pour produire  $\mathbb{R}$ . On écrira alors

```
\newcommand{\R}{\mathbb R}
```

On conseille d'utiliser des raccourcis qui gardent la lisibilité du fichier brut. On ne peut pas utiliser le même raccourcis pour deux commandes différentes, mais `\renewcommand{}{}` permet d'utiliser une commande déjà existante pour autre chose.

# Images

L'utilisation d'images nécessite l'extension `graphicx`. On insère une figure avec la commande `\includegraphics[]{}`. On peut insérer fichiers `.jpg`, `.png`, `.pdf`, etc.

Les figures apparaissent normalement dans l'environnement `figure`, qui permet de rajouter une description avec la commande `\caption{}`.

Pour produire des images qui s'intègrent bien à l'environnement  $\text{\LaTeX}$ , on utilise des logiciels comme `Tikz`, `Metapost`, `Pstricks`, qui ont un langage d'écriture différent, mais suivent le même principe écriture-compilation.

# Bibliographie

Pour avoir une bibliographie, on aura besoin de l'extension `amsrefs`. Elle placée d'habitude à la toute fin d'un document. On peut soit produire un fichier auxiliaire `biblio.bib` avec toutes les références, et l'appeler avec `\bibliography{biblio}`. Chaque entrée bibliographique a une **clé**, avec laquelle on pourra citer le texte par la commande `\cite{clé}`.

Une entrée bibliographique nécessite *auteurs, titre, journal, année, volume, pages*. Le site de l'AMS *MathSciNet* permet de récupérer ces données directement en format `.bib`.

# Liens interactifs

En rajoutant `\label{clé}` après dans un environnement numéroté, comme `theorem`, `equation`, on pourra créer un lien hypertextuel au théorème ou équation souhaité. Cela se fait avec `\ref{}`, ou `\eqref{}` pour les équations.

# Références conseillées

## En ligne :

- <https://fr.wikibooks.org/wiki/LaTeX>
- <https://www.gutenberg.eu.org/> (Groupe francophone des Utilisateurs de TEX, LATEX et logiciels compagnons)
- <https://tex.stackexchange.com/>
- <http://detexify.kirelabs.org/classify.html>