

Notions de base pour Latex

Manuel Paquette-Dupuis

et

Stéphane Surprenant

janvier 2018

Table des matières

1	Installer Latex	4
1.1	MikTex	4
1.2	Interface de programmation	4
1.3	TexStudio	5
2	En-tête d'un document Latex	6
3	Corps	8
3.1	Quelques commandes utiles	8
3.2	Environnements importants	10
4	Bibliographie	14

Introduction

Latex est un langage qui permet de programmer la mise en forme de textes. Le fait de pouvoir automatiser la mise en forme permet d'assurer une cohérence dans la présentation, en plus de sauver beaucoup de temps en cas d'ajouts, de retraits ou de modification du contenu. Imaginer générer 30 fois un tableau de résultats empiriques et devoir, à chaque fois, l'importer sur un logiciel comme Word et l'ajuster manuellement à chaque fois pour en harmoniser la présentation avec le reste du document. Avec Latex, une fois la mise en page programmée, il suffirait simplement de lancer le code après avoir écrasé l'ancien tableau en sauvgardant la nouvelle version.

Ce document a pour but de vous procurer le stricte minimum pour pouvoir apprendre à vous servir de Latex. Il y a beaucoup de détails pour lesquels votre meilleur ami sera Google. Pour la rédaction d'une thèse ou d'un mémoire, il serait incroyablement ridicule de ne pas s'en servir, d'autant que les normes de l'ESG sont déjà programmées pour le corps du document, comme pour la bibliographie et que les logiciels statistiques comme EViews, Stata, Matlab, R ou encore SAS ont tous des fonctions permettant d'exporter des figures et des tableaux directement utilisables par Latex. Aussi, pour ceux et celles qui n'ont pas encore terminé le cours ECO5072 (Activité de synthèse), vous allez rapidement récupérer l'investissement initial d'apprendre Latex en rédigeant les deux rapports exigés.

Le document est divisé comme suit : la première partie couvre brièvement l'installation, la seconde partie traite de l'en-tête des documents `.tex`, la troisième partie passe en revue un ensemble de fonctions et notions très utilisées et la quatrième partie explique comment utiliser *natbib* conjointement avec le style *apalike-uqam* pour automatiser vos besoins bibliographiques.

1 Installer Latex

Il vous faut deux logiciels pour utiliser Latex : (1) un logiciel pour interpréter le langage et, (2), une interface de travail pour coder et lancer les codes.

1.1 MikTeX

Un logiciel permettant à l'ordinateur de «lire» le code Latex. Voici le lien pour le téléchargement : <https://miktex.org/download>. Il vous suffit de suivre les instructions pour installer ce logiciel sur votre ordinateur.

1.2 Interface de programmation

Ici, il y a plusieurs options. Personnellement, nous utilisons **TexStudio** : <https://www.texstudio.org/>. Certaines de mes collègues utilisent plutôt *TexMaker* : http://www.xmlmath.net/texmaker/index_fr.html. Il suffit d'en choisir un et de l'installer en suivant les instructions.

1.3 TexStudio

Une fois que vous avez tout installer, vous pouvez ouvrir votre logiciel de programmation Latex. Si vous ouvrez TexStudio, vous allez avoir quelque chose comme ceci :

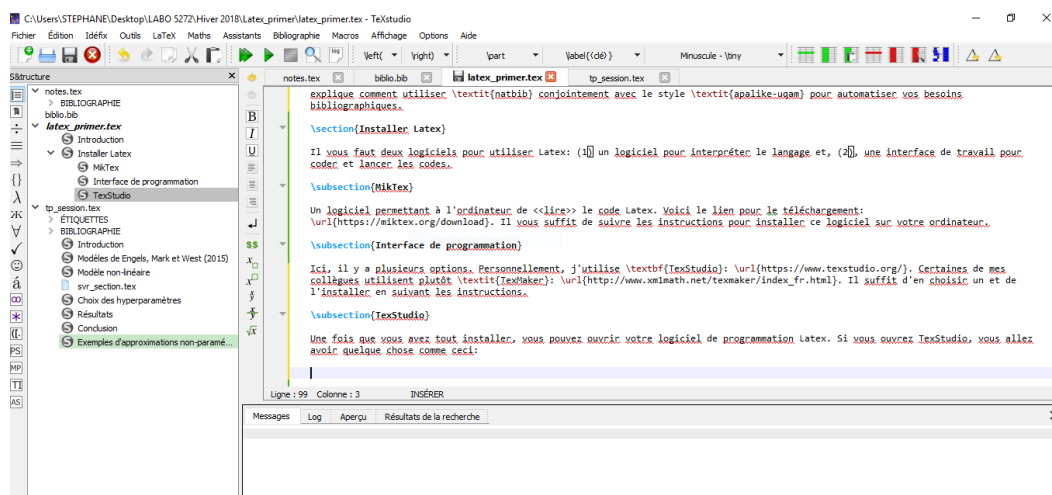


FIGURE 1 – Capture d'écran de TexStudio

À gauche se trouve les documents `.tex` ouverts dans TexStudio. Les sections et les documents importés sont aussi visibles pour chaque fichier `.tex` utilisé. À droite se trouve le script, c'est-à-dire l'endroit où les commandes sont entrées. La double flèche verte en haut permet d'exécuter le code et de visualiser le document. En bas se trouve une console où des messages, dont des erreurs, s'affichent.

2 En-tête d'un document Latex

L'en-tête d'un fichier Latex commence toujours par la commande

`\documentclass[options]{type_de_document}`. Pour un mémoire ou une thèse à l'UQÀM, il existe des styles de documents programmés pour une mise en page conforme aux normes de l'université. Pour des travaux, il vous suffit toutefois d'utiliser un style simple comme *article* et d'ajuster les marges.

L'en-tête contient aussi d'autres éléments essentiels. D'abord, il faut importer les *packages* utilisés, un peu comme dans R. Voici quelques *packages* utiles :

```
\usepackage[french]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[hyphens]{url}
% French

\usepackage{natbib}
% Bibliography

\usepackage[hyphens]{url}
% Allowing URLs to be divided between lines

\usepackage{tabu}
\usepackage{array}      % Matrices (entre autres)
\usepackage{graphicx}

\usepackage{amsmath}    % Symboles
\usepackage{amsfonts}
\usepackage{amssymb}

\usepackage{float}
\usepackage{booktabs}
```

```

\usepackage{adjustbox} % Compresser un tableau
\usepackage{caption} % Titres pour figures et tableaux
% Tables and figures

\usepackage[margin=1.5in]{geometry} % Marges

\usepackage{setspace} % Fonction double et simple interligne

```

En plus des *packages*, l'en-tête contient aussi des définitions de fonctions. Il vous est possible de modifier des fonctions existantes avec `\renewcommand`. Par exemple, `\renewcommand{\tablename}{Tableau}` change le nom des titres plaqués par `\caption` aux tableaux de l'anglais *table* au français, tableau.

Il est aussi possible de créer de nouvelles fonctions avec `\newcommand`. Par exemple, vous pouvez créer une fonction pour construire facilement des dérivée partielle comme ceci :

```

\newcommand{\parder}[2]{\frac{\partial #1}{\partial #2}}

```

Si ceci est ajouter à l'en-tête (en plus des *packages mathématiques adéquats*), taper `\parder{x}{y}` dans un environnement approprié dans le corps du document renvoie $\frac{\partial x}{\partial y}$. Si vous avez beaucoup de dérivées à taper, ceci allège considérablement le code et sauve un peu de temps.

Une dernière chose souvent présent dans l'en-tête est de l'information pour une page de présentation. Par exemple,

```
\title{titre}  
\author{nom}  
\date{date}
```

Fait en sorte que la fonction `\maketitle` appelée dans le corps du texte produise la page titre que vous voyez au début de ce texte.

3 Corps

Le corps d'un document Latex est en fait un type d'environnement. Tout environnement dans Latex débute par la mention `\begin{}` et se termine par `\end{}`. Ici, l'environnement est `document`.

3.1 Quelques commandes utiles

Segmenter le texte

Les commandes `\chapter{titre}`, `\section{titre}`, `\subsection{titre}` permettent de segmenter le texte tout en conservant une trace des titres, de la hiérarchie des titres et de leur ordre. Cet ordre est repris lorsqu'on désire insérer une table des matières. La commande `\appendix` affecte la numérotation des titres : après cette commande, les titres sont numérotés par des lettres.

Sauts et ajustements manuels de l'espacement

`\\` indique un saut de ligne, `\hspace{}` et `\vspace{}` permet d'ajouter des espacements verticaux ou horizontaux – ou d'enlever de l'espace, si une mesure négative est utilisée. De même `\\[]` permet un saut de ligne et d'enlever de

l'espace si la valeur indiquée entre crochet (par exemple, `\[-5ex]`) est négative.

Commandes particulières

« `\` » permet souvent d'utiliser des caractères fonctionnels de façon textuelle. Par exemple, `{ }` encadre souvent des arguments pour des commandes Latex. Pour se servir de `{ }` textuellement, il faut ajouter « `\` ». De même, `%` indique le début d'un commentaire. Si on désire l'utiliser textuellement, il suffit d'ajouter « `\` » devant.

Exemples : `\{ }`, `\[]`, `\%`.

Des commandes comme `$$` et `\[]` permettent d'écrire sur une ligne de texte des expressions habituellement utilisés dans des environnements mathématiques. Par exemple,

```
\begin{equation}
  y_t = X_t \textbf{\textit{beta}} + \textbf{\textit{epsilon}}_t
\end{equation}

$y_t = X_t \textbf{\textit{beta}} + \textbf{\textit{epsilon}}_t$
```

Sont deux façons de produire une équation. Toutefois, la première crée l'équation dans l'environnement *equation*. Par défaut, ceci la met à part et ajoute une numérotation et, la seconde, inscrit l'équation directement dans une ligne de texte :

$$y_t = X_t\beta + \epsilon_t \tag{1}$$

La seconde version permet d'inscrire $y_t = X_t\beta + \epsilon_t$ au beau milieu d'une phrase.

FIGURE 2 – Deux façons d'écrire des équations

Il existe typiquement des commandes pour écrire dans le texte des expressions qui seraient normalement illisibles pour Latex. Chercher les commandes dites *in line* correspondant à l'environnement de votre choix.

3.2 Environnements importants

Figures

Dans Latex, certaines commandes ne sont lisibles que dans certains environnements. Par exemple, les expressions pour créer des indices, des fractions ou des lettre grecques sont lisibles dans les environnements *equation* ou encore *align*, mais pas dans l'environnement *document* directement. Certains environnements permettent plusieurs lignes (par exemple, *align*), alors que d'autres non (par exemple, *equation*). Vérifiez que les expressions que vous utilisez sont cohérentes avec l'environnement dans lequel elles se trouvent si jamais des erreurs surviennent.

Un exemple d'environnement *figure*

```
\begin{figure}[H]
  \begin{subfigure}{.5\textwidth}
    \includegraphics[width=2.3in]{../50_figures/f_13.png}
  \end{subfigure}%
\end{figure}
```

```

\begin{subfigure}{.5\textwidth}
    \includegraphics[width=2.3in]{../50_figures/f_14.png}
\end{subfigure}
\begin{subfigure}{.5\textwidth}
    \includegraphics[width=2.3in]{../50_figures/f_15.png}
\end{subfigure}%
\begin{subfigure}{.5\textwidth}
    \includegraphics[width=2.3in]{../50_figures/f_16.png}
\end{subfigure}
\begin{subfigure}{.5\textwidth}
    \includegraphics[width=2.3in]{../50_figures/f_17.png}
\end{subfigure}\[-1em]
\caption{Titre de la figure}
\begin{footnotesize}
    Note: Quelques notes sous la figures.
\end{footnotesize}
\end{figure}

```

L'environnement *subfigure* permet de combiner plusieurs figures dans une seule. `[width=2.3in]` est une option permettant de fixer les dimensions des petites sous-figures. `\caption` définit un titre, `\includegraphics` importe un fichier en tant qu'image. Latex cherche par défaut dans le répertoire dans lequel se trouve le document `.tex` qui est exécuté. Si vous travaillez à partir d'un fichier de travail dans lequel se trouve un répertoire « 30_latex » et un répertoire « 50_figures », respectivement pour le document `.tex` et les figures, il est utile d'utiliser ce qui s'appelle des **chemins relatifs**.

Pour revenir en arrière d'un dossier, deux points (..) sont entrés avant la première séparation. Dans notre cas, si le dossier « Travail » contient les deux dossiers susmentionnés, `{../50_figures/f_16.png}` indique qu'il y a un répertoire nommé

« 50_figures » dans « Travail » (soit un niveau au-dessus de « 30_latex ») et que le fichier `f_16.png` s’y trouve. Notez que dans Latex, les chemins sont indiqués par des *forward slash* et non des *backslash*. Notez que l’importations ne se limite pas aux figures. Vous pouvez importer des fichiers `.tex` avec la commande `\input{}`. Les tableaux extraits de logiciels statiques sont souvent sauvegardés en format `.tex` pour permettre leur importation de cette façon.

L’environnement `footnotesize` réduit la taille du texte. Les % indique comment organiser les sous-figures (soit une à côté de l’autre plutôt qu’une au-dessus de l’autre).

Exemple de l’environnement *Table*

```
\begin{table}[H]
\begin{center}
\caption{Tests de saisonnalit\'{e} Kruskal–Wallis} \label{kw_test} \small\small
\begin{tabular}{lcccccc} \toprule\toprule
& \multicolumn{3}{c}{\$\Delta \ln A_t^{(2)}\$} & \multicolumn{3}{c}{\$\Delta \ln Y_t\$} \\
& KW & DL & Valeur p & KW & DL & Valeur p \\
Canada (T) & 3,0494 & 11 & 0,9901 & 431,7969 & 11 & 0 \\
Canada (B) & 4,9891 & 11 & 0,9317 & 451,9053 & 11 & 0 \\
Canada (S) & 1,6952 & 11 & 0,9993 & 384,0472 & 11 & 0 \\
Terre–Neuve (T) & 3,1991 & 11 & 0,9878 & 405,7299 & 11 & 0 \\
Terre–Neuve (B) & 9,6117 & 11 & 0,5656 & 379,6131 & 11 & 0 \\
Terre–Neuve (S) & 1,5735 & 11 & 0,9995 & 352,1401 & 11 & 0 \\
\bottomrule\bottomrule
\end{tabular}
\end{center}
\begin{footnotesize}
```

```

\flushleft
Notes :
\end{footnotesize}
\end{table}
\normalsize

```

L’environnement *center* centre le texte, la commande `\multicolumn{a}{b}{c}` permet à l’expression *c* de couvrir *a* colonnes de façon *b*. Les justifications pour *b* sont centrée (c), à gauche (l) et à droite (r). Le même principe s’applique à l’argument de l’environnement *tabular*. Les expressions & séparent les colonnes et \\ séparent les lignes dans l’environnement *tabular*. Les commandes `\bottomrule`, `\toprule`, `\midrule` créent des traits horizontaux. `\small` et `\normalsize` ajustent la taille du texte. Les expressions comme `\{e}` sont des commandes natives de Latex pour créer des accents – dans ce cas-ci, le résultat est «é». Ce n’est pas nécessaire si nous avons les bons *packages*.

Finalement, l’ajout de `\label{kw_test}` permet de créer un objet de référence. Avec ceci, à tout endroit dans le texte où se trouve `\ref{kw_test}`, le numéro du tableau sera reproduit. De cette façon, si on décide de changer l’ordre des tableaux, le texte s’ajuste en conséquence et nous n’avons pas à chercher des chiffres entrés manuellement pour les changer. Cette commande fonctionne aussi pour les figures, les sections, les sous-sections, les chapitres et les équations.

4 Bibliographie