

Rédiger un rapport en Latex

Joseph Salmon

Table des matières

1	Introduction	1
2	Comment démarrer ?	2
2.1	Logiciels	2
2.2	Format de fichier et compilation	2
2.3	Les Packages	2
2.4	Formules Mathématiques et Théorèmes	3
2.5	Organisation du document	4
3	Opérations spéciales	5
3.1	Insérer une image	5
3.2	Liens hypertextes	6
3.3	Insérer un algorithme	7
4	Bibliographie	7
A	Démonstrations	9
B	Programmes	9
C	Fin	9

1 Introduction

Voici un bref aperçu du type de document que l'on peut rédiger avec Latex. Ce texte présente seulement la classe nommée "article" (définie par la commande `\usepackage{article}`). C'est cette classe qui vous permettra d'écrire un rapport pour votre projet.

Pour plus d'information et approfondir vos connaissances en Latex, on pourra également consulter "[The not so Short Introduction to LaTeX2e](#)" de Tobi Oetiker. Pour les non anglophiles, la version en français est disponible [ici](#). Le site internet [FAQ LaTeX de l'équipe Grappa](#) est lui aussi plein de ressource et répond à la plupart des questions courantes. Je

conseil aussi le livre sous licence "Art Libre" de Vincent Lozano [Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander](#).

2 Comment démarrer ?

2.1 Logiciels

Tout d’abord, il faut disposer d’un éditeur de texte et d’un compilateur.

- Sous Mac : installer MacTex (éditeur + compilateur). L’éditeur s’appelle Texshop.
- Sous Linux : Latex est installé de base, il vous faut juste un bon éditeur (VSCode, Sublime Text, Kile, Atom, etc.).
- Sous Windows : installer Miktex d’abord (chercher avec un moteur de recherche) puis Texniccenter ou Winshell par exemple.

2.2 Format de fichier et compilation

En Latex, on écrit un code “barbare” qui sera compiler par un programme. Sur vos terminaux c’est Kile qui compilera : il faut aller dans l’onglet “**build**” ou “**construire**”, puis cliquer sur “**compile**”. On choisit alors le type de compilation (Latex, pdfLatex, bibtex pour la bibliographie ...). Si en sortie on veut un fichier ps : il faut faire “**tex to dvi**” puis “**dvi to ps**” Si en sortie on veut un fichier pdf : il faut faire “**pdfLatex**”. (mieux pour afficher des documents avec des hyperliens et des images de formats standards).

Le fichier Latex du document que vous êtes en train de lire est disponible ici : [ceci est un lien hypertexte](#). Son extension est .tex. Une fois le fichier compilé, on obtiendra un “joli” document en format .pdf ou bien .ps. Pour visualiser le résultat il suffit d’aller dans l’onglet “**build**” (ou construire), choisir “**view**” (ou voir) puis de sélectionner “**Viewpdf**” (si l’on a fait pdfLatex).

Ceci fait, on choisit la classe de document que l’on veut créer. Pour les projets, la classe “**article**” est recommandée (les élèves motivés qui voudraient écrire un livre entier pourront utiliser la classe “**book**”). Le document commence donc par la commande :

```
\documentclass[a4paper,12pt]{article}
```

Les options d’une fonction Latex sont toujours entre crochets. Ici elles déterminent le format (A4) du document, et définissent la taille de la police (12pt).

2.3 Les Packages

En dessous du choix de la classe on donne la définition des packages que l’on veut utiliser. Les packages sont des bibliothèques utilisées pour des fonctions avancées, que l’on doit charger. Par exemple la commande `\usepackage{amsmath}` charge le package “**amsmath**”, qui est utile pour écrire des symboles mathématiques. Le package “**babel**” est lui utilisé

pour écrire des accents (du moins avec l’option “**french**”). En effet, les anglophones n’en ont pas besoin, donc initialement Latex ne gère pas les accents. L’intérêt du package est donc de pallier ce manque.

Les packages sont des fichiers au format .sty. Ils sont pour la plupart déjà installés sur votre ordinateur (par Miktex ou Texshop). Par contre, il se peut que certains packages ne soient pas disponibles. Pour les rendre utilisables pour la compilation il suffit de télécharger le fichier “***.sty” sur internet (on les trouvera sur le site du [CTAN](#)) et de le placer dans le dossier où vous éditez votre fichier .tex.

2.4 Formules Mathématiques et Théorèmes

On peut définir des environnements comme des théorèmes, des propositions, etc. par les commandes :

```
\newtheorem{theorem}{Theorem}[section]
\newtheorem{prop}{theorem}[Proposition]
```

Cela donne :

Théorème 2.1. *L’ensemble des nombres premiers est infini.*

Les théorèmes sont ainsi numérotés par Latex de manière automatique.

Théorème 2.2. *L’ensemble des nombres premiers congrus à 1 modulo 4 est infini.*

Démonstration. ici on ferait la démonstration ... mais je vous la laisse en exercice. □

Une fois les packages définis, on peut créer des raccourcis pour les quantités qui reviennent souvent ou des fonctions. Ici on définit un raccourcis pour la commande qui permet de créer une fonction :

```
\newcommand{\nc}{\newcommand}
```

Remarque 2.1. Toutes les fonctions (au sens informatique du terme) connues par Latex commencent par \ (backslash) comme par exemple \sum qui permet d’afficher le symbole de sommation $\sum_{i=1}^4$.

Pour les formules Mathématiques, on doit les mettre entre deux signes \$ pour qu’elles soit interprétées comme des commandes spéciales et non comme du texte. Pour écrire le signe de sommation ci-dessus on écrit donc $\sum_{i=1}^4$ dans le fichier .tex. Si l’on veut afficher des calculs sur une ligne à part on peut par exemple écrire

$$Sf = \sum_{k \in \mathbb{Z}} \hat{f}[k] e^{i2\pi kt/T}$$

en mettant cette fois les formules entre `\[` et `\]`. On peut aussi utiliser un environnement qui commence par la commande `\begin{blabla}` et finit par la commande `\end{blabla}`. Si `blabla=align`, on peut aligner des calculs facilement :

$$\mathcal{S}f = \sum_{k \in \mathbb{Z}} \hat{f}[k] e^{i2\pi kt/T} \quad (1)$$

$$= \sum_{k \in \mathbb{Z}} \hat{f}[k] e^{i2\pi kt/T} \quad (2)$$

Remarque 2.2. On notera que la numérotation s’est déclenchée toute seule pour les formules. On peut leur donner des noms avec la commande `label`, dont la syntaxe est `\{label\}{nomformule}`. On s’y réfère plus tard dans le document avec la commande `\ref{nomformule}`. Ainsi l’ordre des équations peut être changé sans avoir besoin de renuméroter tout un document. Notre première équation est donc 1. Si l’on ne veut pas numéroter une équation, on met une étoile à la fin de l’environnement (écrire `align*` à la place d’`align` par exemple).

Pour un grand panel de symboles mathématiques, il suffit de regarder les raccourcis fournis par l’éditeur que vous utilisez ou de chercher sur Internet un lexique. Voici néanmoins quelques commandes : pour aller à la ligne `\\` , pour une nouvelle feuille `\newpage`, pour les trois points `\ldots` . . .

On renvoie à “[The Comprehensive Latex Symbol List](#)” par Scott Pakin pour une liste exhaustive de tous les symboles connus.

2.5 Organisation du document

On commence par donner un titre au fichier, le nom de l’auteur et éventuellement la date par les commandes suivantes :

```
\title{Rédiger un rapport en Latex}
\author{Joseph Salmon}
\date{ }
```

Ensuite on commence le corps du document avec `\begin{document}` (ne pas oublier de fermer tout en bas du document par la commande `\end{document}`) . On insère le titre avec `\maketitle` et la table des matières avec `\tableofcontents`. La construction de la table des matières se fait automatiquement. Elle donne l’arborescence du document, dans l’ordre où les chapitres et sous chapitres sont donnés dans le fichier `.tex`.

On crée des chapitres, sous chapitres en écrivant simplement au début d’un chapitre `\section{Le nom du chapitre}` et pour les sous-chapitres `\subsection{Le nom du sous chapitre}`. Latex numérote tout seul les chapitres dans l’ordre où ils se présentent. Si l’on intervertit deux chapitres, à la prochaine compilation les numéros seront automatiquement

échangés. Il faut parfois deux compilations pour afficher correctement la table des matières.

Ceci fait vous pouvez déjà commencer à rédiger un premier document, visuellement correcte avec toutes les formules de mathématiques dont vous avez besoin. Au chapitre suivant on va voir quelques raffinements possibles pour égayer vos documents.

3 Opérations spéciales

3.1 Insérer une image

Après avoir créé des graphiques avec Scilab, Matlab ou tout autre logiciel, vous pouvez avoir envie de les insérer dans votre rapport. Cette opération est plus compliquée qu’avec un éditeur de texte classique. Voici le code pour insérer l’image ci-dessous :

```
\begin{figure} [htb]
\begin{minipage}[b] {0.48\linewidth}
\centering
\centerline {\includegraphics[width=1.44\textwidth]{psnr_sig60.jpg}}
\vspace{0.1cm}
\centerline{(A) un premier graphique}
\medskip
\end{minipage}
\hfill
\begin{minipage}[b]{.48\linewidth}
\centering
\centerline{\includegraphics[width=1.44\textwidth]{psnr_sig60.jpg}}
\vspace{0.1cm}
\centerline{(B) un deuxième graphique}\medskip
\end{minipage}
\caption{Des graphiques encore des graphiques...}
\label{fig :ma_figure}
\end{figure}
```

L’image insérée doit se trouver par défaut dans le même répertoire où se trouve le fichier .tex que vous écrivez. On pourra faire varier les paramètres de taille pour voir comment jouer sur la taille de l’image que l’on insère. Il existe d’autres façons d’insérer des images, mais celle-ci à l’avantage de bien fonctionner ...

En "pdfLatex" (version largement conseillée) on peut insérer des images au format .jpg, .png, .pdf ... En revanche, si l’on compile pour obtenir un fichier .ps avec "**Latex**" il faut insérer des fichiers sous le format (peu connu) .eps. Les commandes en revanche ne diffèrent pas.

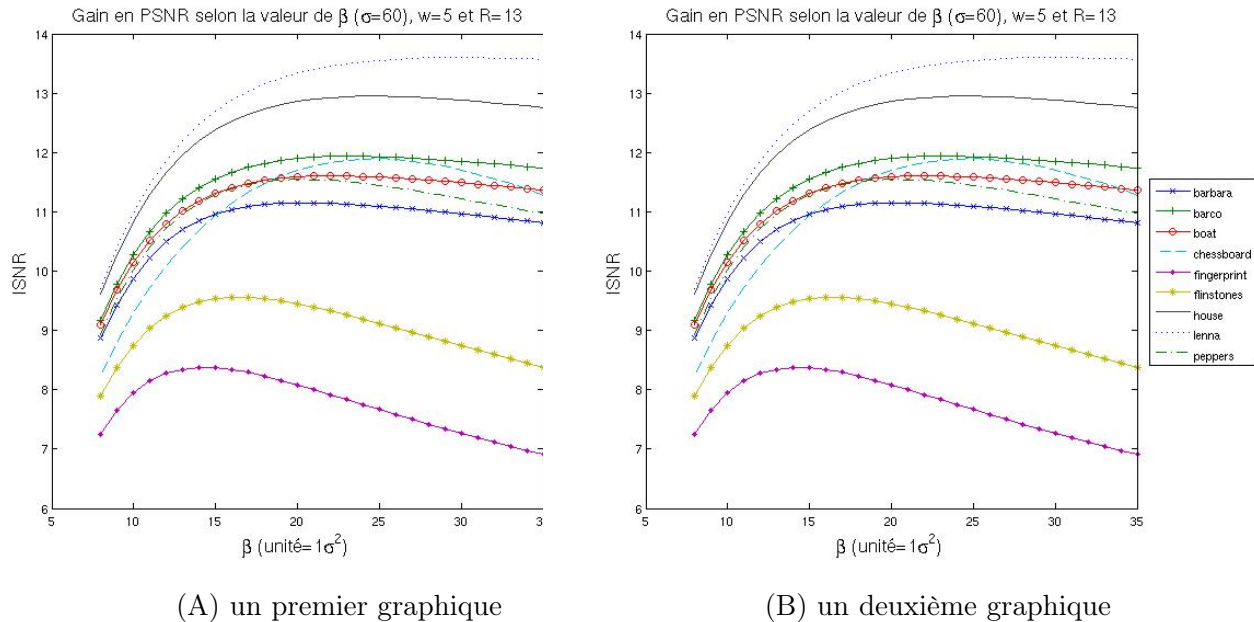


FIGURE 1 – Des graphiques encore des graphiques...

Grâce au label utilisé pour la figure, on peut se référer à celle-ci comme pour les équations, par la commande `\ref{fig :ma_figure}`. Ainsi la première figure est la figure 1.

3.2 Liens hypertextes

Pour ce faire il faut utiliser les packages `pdftex` et `hyperref`. On insère un lien hypertexte par la commande `\url` si l'on veut écrire un lien hypertexte présentant le site cible. Par exemple on peut trouver de l'aide sur latex sur le site officiel : <http://www.ctan.org/> (la commande est `\url{http ://www.ctan.org/}`). Mais on peut aussi le faire avec `\href`, qui permet de définir n'importe quel mot comme lien. Par exemple [là](http://www.ctan.org/) se trouve le site donné ci-dessus (avec la commande : `\href{http ://www.ctan.org/}{là}`). Les références, citations, entrées de la table des matières sont directement hypertextes (c.-à-d qu'on peut cliquer dessus pour aller à l'emplacement correspondant). On pourra utiliser la ligne suivante pour définir divers packages utiles d'un seul coup :

```
\usepackage[pdftex,a4paper,linkcolor=test,citecolor=vertsombre,colorlinks=true,
pagebackref,bookmarks=true, plainpages=true,urlcolor=freeblue]{hyperref}
```

3.3 Insérer un algorithme

Plutôt que de donner un fichier de programmation, il peut être utile d'introduire dans son document un schéma de l'algorithme que l'on a utilisé. Pour cela, on peut choisir les packages suivants :

```
\usepackage{algpseudocode}
\usepackage{algorithm}
\usepackage{algpascal}
```

On peut alors obtenir l'exemple suivant :

```
1: procedure EW-AGGREGATE( $(P(Y)), \sigma$ )
2:    $H = 0$ ;
3:   while  $H < T$  do
4:      $\lambda \leftarrow \lambda + h/T \times L$ ;
5:   end while
6:   return  $\lambda$ 
7: end procedure
```

Les commandes utilisées sont :

```
\allanguage{pseudocode}
\begin{algorithmic}[1]
\Procedure{EW-Aggregate}{ $(P(Y)), \sigma$ }
\State $H=0$ ;
\While{ $H < T$ }
\State $\lambda \leftarrow \lambda + h/T \times L$ ;
\EndWhile\label{euclidendwhile}
\State\textbf{return}  $\lambda$ 
\EndProcedure
\end{algorithmic}
```

Pour plus de détail sur ce package :

www.ctan.org/get/macros/latex/contrib/algorithmicx/algorithmicx.pdf

4 Bibliographie

Pour les références bibliographiques, il faut créer un autre fichier d'extension .bib, qui est l'extension des fichiers “**bibtex**”. On le met également dans le dossier qui contient le fichier .tex que l'on rédige. Le fichier de la bibliographie associée à ce document est disponible [là](#). On présente les articles, livres et autres sites webs de la façon suivante : (on peut trouver également ce type d'information tout rempli sur internet, comme sur le site [MathSciNet](#) par exemple.)

```

@inproceedings{Tsybakov03,
author = {A. B. Tsybakov},
title = {Optimal Rates of Aggregation},
booktitle = {COLT},
year = {2003},
pages = {303-313},
crossref = {DBLP :conf/colt/2003},
bibsourc = {DBLP, http ://dblp.uni-trier.de},
}

```

On doit alors compiler la bibliographie avec la commande `bibtex`. Il faut aller dans l'onglet “**build**” ou “**construire**”, puis cliquer sur “**compile**”. Le type de compilation est alors Bibtex.

La citation apparaît à l'endroit voulu grâce à la commande `\cite{nom citation}`. Tout à la fin du fichier `.pdf`, on trouve alors les références voulues avec tous les détails que l'on a précisés sur le document cité. Pour cela il faut écrire juste avant `\end{document}` les commandes `\bibliography{nom du fichier de la bibliographie}` `\bibliographystyle{alpha}` pour que les citations apparaissent avec des lettres et l'année et non juste des numéros. Par exemple on peut parler de l'article [Tsy03] ou bien du livre [Cat04]. Les articles non cités n'apparaissent pas.

On remarquera qu'à la fin de la bibliographie, les pages où une référence est citée apparaissent sous forme de liens hypertextes. C'est l'utilisation du package “**pagebackref**” qui rend cela possible. Pour plus d'informations sur Bibtex en français : [Le site de Gaël Jaffré](#), [Wikipedia](#), et en anglais : www.bibtex.org

ANNEXE

Enfin avec la commande `\appendix`, on crée un chapitre à part dans lequel la numération est distincte du corps du document.

A Démonstrations

Ici on peut mettre en appendice des calculs trop longs.

B Programmes

Ici on peut mettre les programmes informatiques que l'on a rédigés ...

C Fin

Voilà bonne rédaction !

Références

- [Cat04] O. Catoni. *Statistical learning theory and stochastic optimization*, volume 1851 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2004. Lecture notes from the 31st Summer School on Probability Theory held in Saint-Flour, July 8–25, 2001. 8
- [Tsy03] A. B. Tsybakov. Optimal rates of aggregation. In *COLT*, pages 303–313, 2003. 8