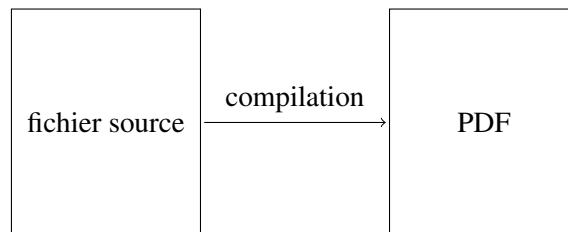


Premiers pas en LaTeX

1.1 Quelques mots sur LaTeX

Le mot LaTeX (prononcé latech) est un jeu de mot basé sur TeX auquel sont rajouté les initiales du concepteur de LaTeX, L. Lamport. TeX lui-même est un logiciel qui a une trentaine d'années (il n'a pas beaucoup évolué depuis) et qui permet, à partir d'un fichier source, de produire, disons, un PDF (il y a d'autres formats de sortie, mais on ne s'y intéressera pas).



Afin d'assurer une homogénéité des fichiers sources (notamment l'encodage en UTF8), on utilisera uniquement TeXmaker qui est disponible sur les 3 plate-formes principales (Windows, Linux, Mac). Il y a une configuration à faire lors de l'installation de TeXmaker pour que le mode de production soit le PDF et que l'encodage soit le bon, voir <http://www.math.jussieu.fr/~goutet/td/install/index.html> pour plus de précisions.

1.2 Structure de base d'un fichier LaTeX

Tout fichier LaTeX commence par la commande

```
\documentclass{article}
```

Noter que la commande commence par \ (anti-slash, qui s'obtient avec les touches AltGr+8, ou Ctrl+Alt+8 sur les machines Windows de Jussieu) et que son nom est `documentclass`, à savoir la classe de document. Son argument est entre une accolade ouvrante { et une accolade fermante }. L'argument de cette commande est `article`; d'autres choix possibles (on sera amené à utiliser aussi `book` et `report`).

L'autre composante absolument obligatoire d'un document LaTeX est

```
\begin{document}
```

```
\end{document}
```

Ce jeu de deux commandes sert à délimiter tout ce qui sera imprimé dans le document (c'est ici qu'on tape le texte qu'on veut voir apparaître dans le PDF). Voici un document LaTeX absolument minimal :

```
\documentclass{article}
\begin{document}
Texte.
\end{document}
```

La partie entre le `\documentclass` et le `\begin{document}` est appelé le préambule ; c'est là qu'on met toutes les définitions et packages qu'on charge.

1.3 Le système de packages de LaTeX

Par défaut, LaTeX comporte peu de fonctionnalités ; cela n'est néanmoins pas grave car LaTeX dispose d'une fonctionnalité de packages qui permet d'étendre les possibilités. Voici la liste des packages qu'on utilisera en standard (voir document vide modèle disponible sur http://www.math.jussieu.fr/~goutet/td/document_vide.tex) :

<code>\documentclass{article}%</code>	autres choix : report, book
<code>\usepackage[utf8]{inputenc}%</code>	encodage du fichier source
<code>\usepackage[T1]{fontenc}%</code>	gestion des accents (pour les pdf)
<code>\usepackage[français]{babel}%</code>	rajouter éventuellement english, greek, etc.
<code>\usepackage{textcomp}%</code>	caractères additionnels
<code>\usepackage{amsmath,amssymb}%</code>	pour les maths
<code>\usepackage{lmodern}%</code>	remplacer éventuellement par txfonts, fourier, etc.
<code>\usepackage[a4paper]{geometry}%</code>	taille correcte du papier
<code>\usepackage{graphicx}%</code>	pour inclure des images
<code>\usepackage{xcolor}%</code>	pour gérer les couleurs
<code>\usepackage{microtype}%</code>	améliorations typographiques
<code>\usepackage{hyperref}%</code>	gestion des hyperliens
<code>\hypersetup{pdfstartview=XYZ}%</code>	zoom par défaut
<code>\begin{document}</code>	
...	
<code>\end{document}</code>	

Regardons plus en détail des packages. Certains packages sont chargés avec des options. Par exemple le package `inputenc` est chargé avec l'option `utf8` pour dire à LaTeX que le document source sera encodé en UTF8 (c'est pour quoi on a réglé l'encodage de TeXmaker sur UTF8) :

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

Ensuite, vient le package `fontenc` qui est chargé avec l'option `T1`. Pour simplifier, disons que ce package permet de s'assurer que les polices de caractères qu'on utilise ont toutes les lettres accentuées dont on aura besoin (si on voulait écrire, par exemple, du polonais, il faudrait utiliser une autre option que `T1`).

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

Ensuite vient le package `babel`. Ce package permet à LaTeX de gérer correctement les langues autres que l'anglais (important pour les coupures de mots, les espacements avant les ponctuations, les caractères spécifiques à une langue, comme les guillemets français, etc.). Pour charger la prise en compte de la langue française, on fait :

```
\usepackage[français]{babel}
```

(Noter l'absence de cédille sur le c de `français` : les options de packages utilisent des caractères non accentués.) Si on voulait écrire de l'anglais, il faudrait utiliser l'option `english` ; pour du grec, l'option `greek`, etc. On peut aussi charger plusieurs langues (ce qui permet de changer au cours du document ; on verra des exemples plus tard) ; dans ce cas, il faut mettre toutes les options voulues au package `babel`, la dernière langue écrite étant celle par défaut. Par exemple, pour un document en français avec quelques passages en anglais, on utiliserait :

```
\usepackage[english,français]{babel}
```

Ensuite, vient le package `textcomp`, qui permet l'accès à certains caractères supplémentaires, non disponibles dans l'encodage T1 utilisé par `fontenc`.

```
\usepackage{textcomp}
```

Ensuite, viennent les packages `amsmath` et `amssymb`, qui sont utiles pour taper les maths (on reviendra dessus à la prochaine séance) :

```
\usepackage{amsmath,amssymb}
```

Ensuite, vient le package `lmodern` qui est le choix d'une police de caractères (le package `fontenc` ne fait que garantir la présence des symboles, il ne choisit pas la police). À la place de `lmodern`, on peut utiliser par exemple `fourier` ; on reviendra sur ce point à une séance ultérieure.

```
\usepackage{lmodern}
```

On charge ensuite le package `geometry`, qui permet de régler la taille du papier et des marges qu'on va utiliser ; on utilise l'option `a4paper` pour spécifier qu'on veut du A4 (par défaut, LaTeX utilise le format de papier américain `letterpaper`).

```
\usepackage[a4paper]{geometry}
```

Ensuite, il y a les trois packages, le package `graphicx`, qui permet d'inclure des images dans le PDF, le package `xcolor` qui permet d'utiliser les couleurs et le package `microtype` qui introduit des améliorations typographiques (toujours charger `microtype` après un package changeant de fonte comme `lmodern` ou `fourier`).

```
\usepackage{graphicx}
\usepackage{xcolor}
\usepackage{microtype}
```

Finalement, on charge le package `hyperref` (toujours à mettre en dernier même si on rajoute des packages supplémentaires). Il permet de gérer les liens, construit les bookmarks du PDF et rend la table des matières interactive (les entrées de la table des matières sont entourées en rouge et lorsqu'on clique dessus, on est amené à la section en question ; le cadre rouge ne s'imprime pas, il est seulement visible à l'écran.)

```
\usepackage{hyperref}
```

On termine par dire à `hyperref` d'utiliser le zoom standard (sans véritable raison, `hyperref` change le zoom à quelque chose de trop petit pour être lu).

```
\hypersetup{pdfstartview=XYZ}
```

Tous les documents produits au cours du semestre devront utiliser le préambule standard qu'on vient de décrire, éventuellement avec des packages supplémentaires au besoin (par exemple, `amsthm`).

1.4 Compilation d'un document avec titre, table des matières et sections

Nous allons maintenant fabriquer notre premier vrai document. Le but est d'arriver à produire le code source correspondant au PDF donné dans les deux pages suivantes.

Évolution de la notion d'intégrale de Cauchy à Riemann

Michel Dupont

8 novembre 1994

Table des matières

1	L'intégrale de Cauchy	1
2	L'intégrale de Riemann	1
2.1	Sommes de Riemann	1
2.2	Définition de l'intégrale de Riemann	2

Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1 L'intégrale de Cauchy

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2 L'intégrale de Riemann

2.1 Sommes de Riemann

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum.

Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2.2 Définition de l'intégrale de Riemann

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Une chose importante à comprendre est que LaTeX est un langage orienté vers la sémantique ; au lieu de dire qu'on va mettre la phrase « Évolution de la notion d'intégrale de Cauchy à Riemann » en gros caractères et centré, on va dire à LaTeX que cette phrase est le titre, puis lui dire de l'afficher. Si ensuite on décide que le titre doit être en gras, il suffira de personnaliser l'apparence du titre.

Analysons la structure du document. Le document comprend tout d'abord un titre, un auteur et une date. Ensuite, il y a une table des matières, puis le corps du document proprement dit. Ce corps de document comprend un titre de section¹ « Introduction » non numéroté (et qui n'est pas dans la table des matières), du texte, puis une section « Intégrale de Cauchy » numérotée, du texte, une section « Intégrale de Riemann » numérotée, une sous-section « Sommes de Riemann » numérotée, du texte, une sous-section « Définition de l'intégrale de Riemann » puis du texte.

Voici les commandes nécessaires pour tous ces éléments. La trois premières commandes, `\title`, `\author` et `\date` sont des commandes à un argument qu'on mettra dans le préambule. Voici un exemple d'utilisation :

```
\title{Titre du document}
\author{Prénom Nom}
\date{7 juin 1986}
```

Si on ne met pas la ligne avec `\date`, la date choisie sera la date du jour de compilation.

Ensuite, après le `\begin{document}`, on utilise `\maketitle` pour imprimer le titre (cette commande ne prend pas d'argument, contrairement aux trois précédentes).

Exercice 1. — En utilisant `\title`, `\author`, `\date` et `\maketitle`, reproduire le titre du document précédent.

Occupons-nous maintenant des sections et de la table des matières. Pour afficher la table des matières, on utilise la commande

```
\tableofcontents
```

Comme `\maketitle`, cette commande ne prend pas d'argument. Ensuite, pour faire une section numérotée, on utilise la commande `\section` dont l'argument est le titre de section ; par exemple :

```
\section{Titre de la section}
```

Si on ne veut pas que la section soit numérotée, on utilise `\section*` ; par exemple :

```
\section*{Titre de la section}
```

Finalement, pour faire une sous-section, on utilise `\subsection` (la syntaxe est la même que pour `\section`).

Exercice 2. — En utilisant `\tableofcontents`, `\section` et `\subsection`, reproduire la table des matières et la structure des sections du document précédent.

Pour finir, il faut taper le texte dans les sections et sous-sections précédentes. Pour cela, on va charger le package `lipsum` puis utiliser la commande `\lipsum` qui affiche le texte en question. Cette commande peut prendre un argument optionnel selon le texte que l'on veut qu'elle affiche. La syntaxe est

```
\lipsum[1]
```

où l'on peut remplacer 1 par un autre nombre.

1. Traditionnellement, on parle plutôt de paragraphe ; néanmoins, pour être en phase avec LaTeX et pour éviter la confusion avec la commande `\paragraph`, on parlera de section

Exercice 3. — En chargeant le package `lipsum` et en utilisant la commande `\lipsum`, finir la reproduction du document précédent.

1.5 Exemple d'utilisation d'un environnement et de commandes simples

On va maintenant s'intéresser à reproduire le bout de texte suivant.

Voici ce que disait Paul Montel :

« Au début de 1941, Henri Lebesgue donna au Collège de France son dernier enseignement annuel. Déjà, le mal qui devait l'emporter quelques mois plus tard était venu s'ajouter aux souffrances morales de la défaite et de l'occupation ennemie. Il ne pouvait guère se déplacer à pied et la ville était dépourvue de moyens de transport en surface. Il dut faire appel à ces appareils tenant de la chaise à porteur et de la bicyclette qui servaient alors pour les malades et put ainsi faire son cours. À ses yeux cela signifiait faire son devoir, discipline plus nécessaire que jamais pour ceux qui, comme lui, avaient foi dans la libération et le relèvement de la Patrie. »

(D'après http://fr.wikipedia.org/wiki/Henri_L%C3%A9on_Lebesgue)

Analysons la structure de ce texte. Il y a une citation, entourée de guillemets, qui est mise en page dans un bloc avec marges à gauche et à droite. Il y a aussi un lien internet. Voici les différentes commandes nécessaires pour reproduire cette mise en page.

Pour mettre en page un lien internet, on utilise la commande `\url` ; par exemple,

```
\url{http://www.google.com}
```

produit <http://www.google.com> où le lien est automatiquement cliquable pour aller à la page web voulue.

Pour taper les guillemets, on utilisera les commandes `\og` et `\fg` (le signifie qu'il faut laisser un espace après la commande). Voici un exemple d'utilisation :

```
\og Science sans conscience n'est que ruine de l'âme.\fg
```

produit « Science sans conscience n'est que ruine de l'âme. »

Finalement, pour créer un bloc de texte avec marges à gauche et à droite, il suffit de mettre ce texte entre un `\begin{quotation}` et un `\end{quotation}`. Par exemple :

```
\begin{quotation}
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.
\end{quotation}
```

produira

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Exercice 4. — Avec les commandes `\url`, `\og` et `\fg` ainsi que l’environnement `{quotation}`, composer le texte donné au début de cette section.

1.6 Définition de commandes personnelles

Considérons le texte suivant :

L’histoire de la théorie de l’intégration est jalonnée de noms célèbres comme Augustin Louis CAUCHY, Bernhard RIEMANN, Henri LEBESGUE ou encore Arnaud DENJOY.

Si on veut reproduire ce texte, il suffit d’utiliser la commande `\textsc` qui prend un argument et le met en petites capitales ; par exemple `\textsc{Cauchy}` donnera CAUCHY.

L’histoire de la théorie de l’intégration est jalonnée de noms célèbres comme Augustin Louis `\textsc{Cauchy}`, Bernhard `\textsc{Riemann}`, Henri `\textsc{Lebesgue}` ou encore Arnaud `\textsc{Denjoy}`.

Le problème de précéder ainsi est qu’il ne sera pas aisé de changer d’avis sur la composition des noms propres si on souhaite plus tard, disons par exemple ne plus afficher les prénoms, ou ne plus mettre les noms en petites capitales ou même rajouter les noms dans l’index.

Une habitude importante à prendre avec LaTeX est de définir des commandes permettant d’avoir accès au sens. Ici, ce que l’on compose, c’est des noms propres ; il est donc parfaitement naturel de vouloir avoir une commande `\nompropre` qui prend deux arguments, le prénom et le nom et les compose de la façon voulue. Dans l’exemple précédent, on voudrait que

```
\nompropre{Augustin Louis}{Cauchy}
```

donne Augustin Louis CAUCHY.

Voyons comment définir de nouvelles commandes en LaTeX. Pour définir une commande qui ne prend pas d’argument, on utilise la syntaxe suivante :

```
\newcommand{\SNCF}{S.N.C.F}
```

Ceci définit une commande `\SNCF` qui imprimera S.N.C.F. Pour avoir des commandes avec argument, il faut spécifier entre crochet, juste après le nom, le nombre d’arguments. On accède ensuite aux différents arguments avec `#1` pour le premier, `#2` pour le deuxième, etc. Par exemple

```
\newcommand{\guillemets}[1]{\og #1\fg{}}
```

crée une commande `\guillemets` à un argument telle que

```
\guillemets{science sans conscience n'est que ruine de l'âme}
```

produise « science sans conscience n’est que ruine de l’âme ».

On veillera à n’utiliser `\newcommand` que dans le préambule. Pour des raisons de lisibilité, il vaut mieux mettre toutes les `\newcommand` après le chargement de tous les packages.

Exercice 5. — Créer la commande à 2 arguments `\nompropre` qui imprime le prénom suivi du nom en petites capitales.

Exercice 6. — Modifier la commande précédente pour qu’elle n’affiche plus que le nom de famille.

1.7 Compilation d'un document avec théorèmes

Après les sections, que l'on a vu dans le § 1.4, un deuxième type d'objets fondamentaux consiste en les théorèmes, corollaires, lemmes et autres propositions ainsi que dans les démonstrations. Ces objets ne sont pas définis par défaut. Une fois qu'on aura défini un théorème, il suffira de taper

```
\begin{theoreme}  
Contenu du théorème.  
\end{theoreme}
```

pour produire le théorème en question.

Le but est de savoir reproduire le document donné à la page suivante.

Analysons ce document. On a des théorèmes, des corollaires, des lemmes, des propositions, des définitions, des remarques et des démonstrations. Ces environnements peuvent être regroupés en quatre catégories :

- les théorèmes, corollaires, lemmes, propositions ont leur nom composé en gras puis leur contenu composé en italique ;
- les définitions ont leur nom composé en gras et leur contenu composé en droit ;
- les remarques ont leur nom composé en italique et leur contenu composé en droit ;
- les démonstration sont comme les remarques, sauf qu'elles impriment un carré à la fin de la démonstration.

La numérotation est automatique et a été choisie de telle façon que le numéro de section soit imprimé avant le numéro du théorème et remis à 1 après chaque changement de section. De plus, la numérotation de tous ces environnements se suit (la première proposition après la définition 1.1 est la proposition 1.2, pas la proposition 1.1).

Pour composer des théorèmes, on a besoin de charger le package `amsthm`. Comme tous les autres packages, il vaut mieux le charger avant `hyperref` ; il vaut mieux également le charger avant `lmodern` car si jamais on change ce package pour un autre (par exemple `txfonts`), il peut y avoir des incompatibilités. Le plus simple est de le rajouter à la ligne où il y a déjà `amsmath` et `amssymb` :

```
\usepackage{amsmath,amssymb,amsthm}
```

Une fois fait, la définition d'un théorème se fait en deux étapes. Il faut d'abord sélectionner un style ; cela se fait avec la commande `\theoremstyle`. Par exemple,

```
\theoremstyle{plain}
```

sélectionnera le style `plain` qui est celui des théorèmes. Il y a aussi le style `definition` pour les définitions et `remark` pour les remarques. Il n'y a pas de styles pour les démonstrations, elles sont produites directement en utilisant

```
\begin{proof}
Bla bla bla bla bla.
\end{proof}
```

Une fois le style sélectionné, on doit définir le théorème proprement dit. La syntaxe est la suivante :

```
\newtheorem{theoreme}{Théorème}[section]
```

Le premier argument est le nom de l'environnement² ; il ne doit pas comporter d'accents. Si on utilise `theoreme`, on tapera plus tard `\begin{theoreme}` ; si on utilise `theo`, il faudra taper `\begin{theo}`. Un nom court est plus rapide à taper, mais rend le fichier source moins lisible. Le deuxième argument est ce que l'on veut qui soit imprimé, ici « Théorème » (avec majuscule et accents). Le troisième argument, entre crochets car optionnel, permet de rajouter le numéro des sections avant le numéro des théorèmes et réinitialise ce numéro lors de chaque changement de section. Si on utilisait la classe de document `book` au lieu de `article`, on voudrait probablement mettre `chapter` ici au lieu de `section`.

Ensuite, on veut définir par exemple un corollaire qui sera numéroté de la même façon que `theoreme`. La syntaxe est :

```
\newtheorem{corollaire}[theoreme]{Corollaire}
```

La seule chose qui change par rapport à ce qu'on a utilisé pour définir l'environnement `theoreme` est l'argument optionnel qui est désormais entre les deux arguments entre accolades. Cela veut dire qu'il faut adopter la même numérotation que `theoreme`.

Tous les autres environnements s'obtiennent de la même façon, sauf pour la remarque qui est non numéroté ; pour elle, on utilise

2. Faire attention, tous les noms ne sont pas permis. Une erreur fréquente est de vouloir appeler une définition `def` ; cela provoque une erreur car c'est un nom réservé. De même, `th` est déjà pris (c'est le caractère θ).

```
\newtheorem*{remarque}{Remarque}
```

Finalement, si on voulait un environnement numéroté indépendamment de tout (« Axiome 1 » puis « Axiome 2 », etc.), on utiliserait

```
\newtheorem{axiome}{Axiome}
```

Exercice 7. — En utilisant les outils qu'on vient de décrire, reproduire le document donné précédemment.