

Le langage \LaTeX

(formatage de documents avec \LaTeX)

– Travaux Dirigés (3) –

Résumé

Le but de ce TD est de vous initier au langage \LaTeX . Vous apprendrez à faire des documents de différents types (articles, rapports, mémoires, livres, présentations, *etc*) et à maîtriser différents aspects de ce langage pour pouvoir améliorer le rendu de votre document.

Table des matières

1	Les bases de \LaTeX	2
1.1	Créer son environnement de travail	3
1.2	Principes du langage	3
1.3	Compilation \LaTeX	5
1.4	Classes documents et paquets	5
1.5	Trouver de l'aide sur \TeX et \LaTeX	6
2	Composition de documents \LaTeX	7
2.1	Fontes	7
2.2	Structure d'un document	9
2.3	Paragraphes	10
2.4	Listes	11
2.5	Tableaux	12
2.6	Images	14
2.7	Références et notes de bas de page	14
2.8	Objets flottants	16
3	Composition avancée de documents \LaTeX	17
3.1	Mathématiques	17
3.2	Bibliographie	18
3.3	Créer des schémas avec \TikZ	20
3.4	Faire une présentation avec Beamer	24
4	Conseils d'écriture et de style	28
4.1	Avant d'écrire!	28
4.2	Faites un plan détaillé et écrivez	28
4.3	Relisez, corrigez et retravaillez	29
4.4	Que doit contenir un état de l'art? Comment faire sa bibliographie?	29
4.5	Comment éviter le plagiat?	29
4.6	Références utiles	30

Avant propos

\LaTeX est un langage servant à décrire et formater des documents puis à produire un rendu de haute qualité typographique. Ce langage est extrêmement répandu dans le monde académique, notamment dans toutes les branches scientifiques, car il permet d'écrire facilement des formules mathématiques, mêmes complexes, avec un rendu bien supérieur à la plupart des autres logiciels de composition de textes (LibreOffice, Windows Word, Apple iWork, *etc*). Il est notamment utilisé pour écrire la plupart des

articles scientifiques, des rapports de projets, des mémoires de fin d'études, des polycopiés de cours et même des livres entiers. Par exemple, cette feuille d'exercices est réalisée grâce à ce langage.

Notez aussi que, certaines communautés dans l'industrie utilisent aussi \LaTeX de manière intensive (essentiellement les laboratoires de recherche privés). Et, il n'est pas exclu, lors d'un de vos stages, que votre encadrant vous demande un rapport ou une présentation qui soit rédigé en \LaTeX .

Une brève histoire de \TeX et \LaTeX

En 1976, Donald Knuth, un chercheur en informatique théorique de l'université de Stanford, publie le troisième tome de sa série '*The Art of Computer Programming*'. D'après son plan original, la série doit compter sept tomes, mais frustré de ne pas pouvoir obtenir un rendu typographique correct avec les logiciels de traitement de texte de l'époque (et étant mécontent du travail de l'imprimeur), il décide d'interrompre momentanément l'écriture de sa série pour se consacrer à la création d'une solution acceptable pour résoudre son problème. Et, c'est en 1978 que sort la première version publique du langage \TeX . Cette première version permettait un rendu typographique de très haute qualité, y compris pour les formules mathématiques, ce qui, à l'époque, était totalement inédit.

Malgré tout, le langage \TeX restait difficile à utiliser par des non-initiés, de par sa complexité et aussi parce que certaines fonctionnalités de base n'étaient pas présentes dans sa version classique. En 1982, Leslie Lamport, un autre chercheur de l'université de Stanford, créa un ensemble d'outils et de macros destinées à rendre l'utilisation de \TeX plus facile. Cet ensemble de macros, qui constituait un langage à part entière en sur-couche à \TeX , fut nommée le "*Lamport \TeX* ", ce qui donna \LaTeX .

À l'heure actuelle, la variation de \LaTeX la plus couramment utilisée est $\text{\LaTeX} 2_{\epsilon}$, basée sur $\text{\TeX} 3$ (dernière version en date du langage \TeX). La première version de $\text{\LaTeX} 2_{\epsilon}$ est sortie dans les années 1990 mais a évolué constamment depuis, en s'enrichissant de fonctionnalités et en corrigeant des bogues.

Quant à l'avenir de \LaTeX , $\text{\LaTeX} 3$ est sorti en 2013. Mais, les changements les plus importants entre les deux versions seront au niveau du cœur du langage et de son moteur de rendu. Pas au niveau du langage de macros qui n'aura que quelques incompatibilités mineures. $\text{\LaTeX} 2_{\epsilon}$ reste donc un langage utile à apprendre si vous désirez composer des documents d'une certaine qualité typographique (ce qui arrivera inévitablement pendant votre scolarité à l'université).

Enfin, le langage \LaTeX ne serait rien sans un '*moteur de rendu*' (aussi appelé '*compilateur \LaTeX* ') qui transforme la description du document en un document final visualisable et imprimable. Si, à l'époque des premières versions, les formats de sortie étaient le DVI (Device independent file format) ou le PostScript, de nos jours la priorité est mise sur le PDF (Portable Document Format). Le compilateur le plus utilisé à l'heure actuelle est donc '*pdftex*' (parfois appelé '*pdf \LaTeX* ') qui génère directement du PDF sans passer par un format intermédiaire, ce qui permet de tirer le meilleur parti de toutes les fonctionnalités du format (insertion d'hyperliens, fontes embarquées, ...).

Attention: Il existe beaucoup de pages Internet qui ne sont pas à jour à propos de \LaTeX . Celles-ci parleront d'anciens formats tels que le DVI et le PostScript. La compatibilité avec les anciennes versions de \LaTeX n'est pas assurée. Il faut donc être vigilant concernant l'origine et la '*fraîcheur*' des informations collectées sur le Web à propos de \LaTeX .

Un dernier point avant de commencer, la prononciation de \TeX et \LaTeX porte souvent à confusion à cause du 'X' final. Il s'agit en fait de la représentation majuscule de la lettre grecque Khi 'χ' (ici en minuscule). \TeX et \LaTeX se prononcent donc, respectivement, *Te'ck* et *Late'ck*.

1 Les bases de \LaTeX

Nous profiterons ici de l'installation \LaTeX faites dans les salles machines de l'université, mais sachez que \LaTeX peut s'installer sur à peu près n'importe quel système d'exploitation un peu courant. Il existe des distributions pour Linux ($\text{\TeX}live$, $te\text{\TeX}$), MacOS X ($Mac\text{\TeX}$, $\text{\TeX}live$, $te\text{\TeX}$), ou MS-Windows ($Mik\text{\TeX}$). Vous pouvez donc forcément l'installer sur votre machine.

1.1 Créer son environnement de travail

Pour ceux d'entre vous qui ne connaissent que des logiciels de traitement de texte, le concept même de \LaTeX risque de sembler un peu étrange. En effet, il ne repose pas sur le WYSIWYG¹ habituel. Au contraire, il y a une séparation nette entre la phase de rédaction du document et la phase de visualisation du rendu. Les deux étapes étant séparées par l'étape dites de *compilation* du document. Le processus de composition d'un document \LaTeX (figure 1) comporte donc trois étapes :

1. Le document est tapé en mode texte dans un **éditeur de texte** (Emacs, Vi(m), gEdit, ...).
2. Le fichier \LaTeX est compilé² par un **moteur de rendu** ou **compilateur** (pdf \TeX) pour produire un PDF.
3. Le document est enfin visualisé avec un **logiciel de visualisation de PDF**³ (evince, okular, ...).

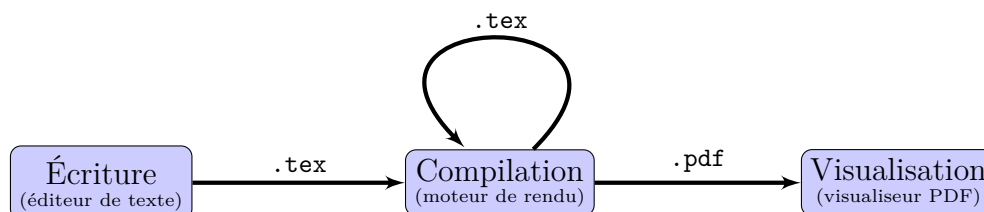


FIGURE 1 – Processus (simplifié) de composition d'un document \LaTeX .

Questions

1. Choisissez un éditeur de texte que vous connaissez bien et qui, si possible, a un mode \LaTeX .
2. Choisissez un visualiseur de fichiers PDF qui est présent dans votre environnement de travail.
3. Ouvrez l'éditeur de texte et le visualiseur que vous avez choisi, ainsi qu'un terminal (pour lancer la compilation) sur votre environnement de travail. Placez les afin qu'ils soient facilement accessibles pendant toute votre session de travail.

1.2 Principes du langage

Par convention, tous les fichiers contenant du \LaTeX (ou du \TeX) ont une extension `.tex`. Ces fichiers sont de simples fichiers textes qui sont eux-mêmes composés de plusieurs parties ayant des rôles bien précis. Comme nous pouvons le voir sur le listing 1 les quatre parties suivantes :

- D'un appel à une *classe de document* qui charge un style global ;
- D'une liste (optionnelle) de *paquets* qui ajoutent des commandes supplémentaires au langage de base ou des fonctionnalités particulières ;
- D'un ensemble de commandes permettant de personnaliser le style global ;
- Et enfin, du contenu du document.

Notez que le caractère `'%` sert à marquer les commentaires en \TeX et en \LaTeX . D'ailleurs, profitons-en pour lister les 10 caractères spéciaux de \LaTeX , nous verrons à quoi ils servent plus tard. Remarquez que si vous désirez les utiliser il faudra les faire précéder d'un `'\'` :

`\$ & \% \# ^ _ \{ \} \sim`

Le langage \LaTeX repose essentiellement sur deux types de constructions : les '*commandes*' et les '*environnements*'. Une '*commande*' peut avoir deux types d'usage. Elle peut être utilisée pour modifier certains paramètres du style, charger des paquets, modifier le rendu d'un mot ou d'un ensemble de mots

1. WYSIWYG : *What You See Is What You Get*

2. Notez que plusieurs compilations sont parfois nécessaires pour générer la table des matières, les références croisées, etc.

3. Préférez un logiciel de visualisation qui rafraîchisse automatiquement la vue du document en cas de modification du fichier, sinon vous devrez le relancer manuellement à chaque compilation.

```

\documentclass[a4paper,10pt]{article} % Appel a une classe de document
\usepackage{amsmath} % Appel a un paquet optionnel

\usepackage[utf8]{inputenc} % Reconnaissance de l'UTF-8
\usepackage[T1]{fontenc} % Chargement de fontes truetype

\usepackage[french]{babel} % Localisation en francais

% Personnalisation du style global
\pagestyle{empty}

\begin{document}
% Contenu du document
Bonjour le monde de \LaTeX !
\end{document}

```

Listing 1 – Document \LaTeX minimal

à l'intérieur d'un paragraphe, *etc.* Alors qu'un '*environnement*' va permettre de définir une zone de texte dans laquelle on va pouvoir redéfinir totalement (ou seulement partiellement) les règles du langage.

Le tableau 1 résume les différentes syntaxes de ces deux éléments du langage. Vous noterez que certaines commandes ou environnements peuvent avoir recours à des options. Mais, il faut aussi savoir que certaines options sont obligatoires (ce qui est paradoxal, mais c'est ainsi).

Type	Syntaxe
Commande simple	<code>\command</code>
Commande avec un argument	<code>\command{arg}</code>
Commande avec deux arguments	<code>\command{arg1}{arg2}</code>
Commande avec une option et un argument	<code>\command[option]{arg}</code>
Commande avec deux options et un argument	<code>\command[option1,option2]{arg}</code>
Environnement ' <i>env</i> '	<code>\begin{env}</code> ... <code>\end{env}</code>
Environnement ' <i>env</i> ' avec une option	<code>\begin[option]{env}</code> ... <code>\end{env}</code>

TABLE 1 – Les commandes et les environnements en \LaTeX .

Attention: Notez que le \LaTeX est un langage de formatage de documents, et qu'à ce titre, il mérite autant de soin que lorsque vous programmez dans des langages de programmation classiques (C, Java, Python, ...). Il est donc important d'indenter votre code, d'y insérer des commentaires, de donner des noms compréhensibles à vos variables et macros, de respecter les 80 colonnes, *etc.* Bref, la lisibilité du code \LaTeX doit être mise en avant. Ceci vous aidera beaucoup lorsque vous voudrez récupérer une partie de votre code et l'utiliser dans un autre document ou si jamais vous rencontrez un bogue ou encore si vous écrivez un document à plusieurs.

Questions

1. Identifiez les commandes et environnements avec leurs arguments et options dans le listing 1.
2. Recopiez le listing 1 dans un fichier `mydocument.tex`.

1.3 Compilation L^AT_EX

Le travail du moteur de rendu (que nous appellerons un “*compilateur* L^AT_EX”) est de produire un document qui possède le meilleur rendu typographique possible. Pour cela, il faut savoir que chaque classe de document laisse une certaine latitude au compilateur pour effectuer de petits choix locaux (insertion de sauts de page, de sauts de ligne, d’espacement entre les paragraphes, ou encore différents positionnement des objets flottants). En outre, le compilateur possède une fonction d’évaluation qui lui permet d’attribuer une note de ‘*beauté*’ à chaque document obtenu. Le but du compilateur sera donc de trouver, par essais-erreurs, l’ensemble des choix qui lui permettront d’obtenir rapidement la plus haute note possible.

Évidemment, en L^AT_EX il est toujours possible de faire pencher la balance plutôt en faveur d’un choix qu’un autre (forcer un saut de page, déplacer un objet flottant avant un paragraphe, *etc*). Mais, le compilateur aura toujours le dernier mot. Quoiqu’il en soit, apprenez à accepter les choix du compilateur le plus souvent possible et à ne pas vous battre contre lui. C’est un combat qui est souvent vain.

Au passage, faire passer le rendu par une compilation est une différence fondamentale entre les logiciels de traitement de textes et L^AT_EX. Les premiers ont une vision seulement locale de chacun des choix faits, alors que L^AT_EX nécessite une vision globale du document pour faire ses choix. La cohérence typographique du document en est donc renforcée et, avec elle, l’esthétique. L’inconvénient majeur de cette méthode est qu’elle rend plus difficile une visualisation en temps-réel du document, vu la quantité de traitements qui est nécessaire pour obtenir le rendu final.

Enfin, lors d’une compilation, vous noterez aussi qu’un certain nombre de fichiers intermédiaires sont produits par le compilateur (‘.aux’, ‘.bbl’, ‘.toc’, ...). Ils servent essentiellement d’étapes intermédiaires avant de produire le document final. Ces fichiers contiennent des informations sur les références croisées, les pages dans lesquelles apparaissent les objets flottants ainsi que sur les différents choix qui ont été fait par le compilateur. Ce sont ces fichiers qui sont utilisés comme point de départ lorsqu’il est nécessaire d’avoir plusieurs compilations pour obtenir le document final. Typiquement, lorsque vous avez des références croisées, ou bibliographique, la première compilation remplacera les numéros attendus par des ‘?’ pour indiquer qu’il faut une deuxième compilation pour les remplacer par les numéros auxquels ils correspondent.

Usage	Commande
Affiche l’aide	?
Affiche une explication détaillée de l’erreur	h
Force la compilation à continuer	<Enter>
Affiche les messages d’erreurs suivants	s
Poursuis la compilation sans arrêt	r
Poursuis la compilation sans afficher les messages (<i>quiet</i>)	q
Insertion d’un texte (par exemple, une balise oubliée)	i
Éditer le fichier source concerné	e
Abandon de la compilation	x
Ignore les n prochaines erreurs (n entre 1 et 9)	1-9

TABLE 2 – Les commandes internes du compilateur.

Questions

1. Compilez `mydocument.tex` en faisant ‘`pdflatex mydocument.tex`’ et visualisez le document PDF qui en résulte.
2. Provoquez une erreur de compilation en modifiant le fichier `mydocument.tex`, puis essayez les diverses commandes qui permettent d’interagir avec le compilateur (voir table 2).
3. Identifiez les fichiers intermédiaires qui sont produits par la compilation et regardez leur contenu.

1.4 Classes documents et paquets

Comme vous l’avez vu dans le listing 1, le document L^AT_EX commence par appeler une classe de document : `\documentclass{article}`. La classe de document ‘`article`’ contient un ensemble de réglages de

style par défaut qui permettra de produire un document du type ‘article’. La table 3 récapitule toutes les classes de documents de base, plus quelques-unes qui sont non standard mais très couramment utilisées. Évidemment, il est très courant de partir d’un des styles par défaut et de la personnaliser ensuite. Mais, attendez d’être plus habitué à L^AT_EX avant d’essayer ce genre de chose.

Usage	Classe
Article scientifique	<code>article</code>
Livre	<code>book</code>
Lettre	<code>letter</code>
Rapport	<code>report</code>
Présentation	<code>slide</code>
Présentation (avancée)	<code>beamer*</code>
Mémoire (de thèse)	<code>memoir*</code>

Usage	Option
Taille de la fonte	<code>10pt*</code> , <code>11pt</code> , <code>12pt</code>
Format de la feuille	<code>a4paper*</code> , <code>letterpaper</code>
Nombre de colonnes	<code>onecolumn*</code> , <code>twocolumn</code>
Passe en mode brouillon	<code>draft</code>

TABLE 3 – Les paramètres de la commande `\documentclass`.

À gauche : Les classes de documents les plus courantes (* marque les classes non-standards).

À droite : Quelques options utiles de la commande (* marque les options par défaut).

L’autre point fort de L^AT_EX vient aussi de son système de paquets (*‘packages’*) qui permet d’enrichir facilement le langage pour des usages très spécifiques. Par exemple, dans notre fichier minimal (listing 1), on charge le paquet de reconnaissance de l’UTF-8 ainsi qu’un paquet de localisation (`babel`) en lui précisant de charger le Français.

Il existe un très grand nombre de paquets qui couvrent presque toutes vos envies de fonctionnalités (passées, présentes et futures). Il suffit de savoir les trouver et de lire attentivement leur documentation⁴. Nous verrons, d’ailleurs, plusieurs autres paquets indispensables dans la suite de ce TD et nous apprendrons à nous en servir.

1.5 Trouver de l’aide sur T_EX et L^AT_EX

La littérature qui existe sur L^AT_EX est extrêmement abondante, mais tout n’est pas forcément à lire. Voici une liste de sources relativement ‘sûres’ de documentation :

- Wikibook L^AT_EX : <http://en.wikibooks.org/wiki/LaTeX>
- ‘The (Not So) Short Introduction to L^AT_EX2_ε’, par Tobi OETIKER : <http://tobi.oetiker.ch/lshort/lshort.pdf>
- Apprend L^AT_EX, par Marc BAUDOUIN : http://www.babafou.eu.org/Apprends_LaTeX/Apprends_LaTeX.pdf
- Tout ce que vous avez toujours voulu savoir sur L^AT_EX, par Vincent LOZANO : <http://lozzone.free.fr/latex/guide-local.pdf>
- L^AT_EX and Friends, par M. R. C. VAN DOGEN : <http://csweb.ucc.ie/~dongen/LAF/LAF.html>
- L^AT_EX Tutorials : A primer, par Indian TeX Users Group, 2003. <http://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf>
- Conseils pour bien taper un document avec L^AT_EX, 2009-2010. http://www.math.ens.fr/~millien/tdlatex/conseils_latex.pdf
- Petites leçons de typographie, par Jacques ANDRÉ : <http://jacques-andre.fr/faqtypo/lessons.pdf>

Quelques sites liés à T_EX et L^AT_EX :

- T_EX.stackexchange : <http://tex.stackexchange.com/>
- T_EXBlog : <http://texblog.org/>
- L^AT_EX-project : <http://www.latex-project.org/>
- T_EX Users Group : <http://www.tug.org/>
- Comprehensive T_EX Archive Network : <http://www.ctan.org/>

4. Voir sur <http://www.ctan.org/>

2 Composition de documents L^AT_EX

Écrire un document avec L^AT_EX se fait en plusieurs phases. La plupart du temps on commence par définir une structure du document en découpant le texte en chapitres, sections, sous-sections, *etc.* Puis, on rédige les différentes parties du document. Enfin, on passe à la mise en page en s'intéressant à son rendu final et en retouchant et corrigeant le texte au passage.

2.1 Fontes

Le choix d'une fonte dépend de votre goût mais aussi de facteurs techniques. Par exemple, toutes les fontes ne supportent pas le mode mathématique. Cela n'a pas beaucoup d'importance si votre document ne contient pas de formules mathématiques, mais si c'est le cas, le mode mathématique basculera automatiquement sur le jeu de fontes par défaut pour représenter les formules et cela risque de trancher par rapport au reste du document. Il faut donc essayer de choisir votre jeu de fontes en fonction de l'effet que vous désirez et de quelques caractéristiques techniques⁵. En L^AT_EX, les fontes sont caractérisées par six attributs :

- L'**encodage** : Les deux principaux sont l'**OT1** (*T_EX text*) et le **T1** (*T_EX extended text*). Le plus souvent vous aurez besoin du T1 car vous utiliserez des caractères accentués⁶.
- Le **jeu de fontes** : Le nom de la collection de caractères que vous utilisez. Notez que tous n'ont pas le support pour le mode mathématique (voir table 4).
- La **famille** : Correspond aux caractéristiques typographiques de base comme la présence de "pattes" à l'extrémité des lettres (Sérf) ou pas (Sans Sérf).
En L^AT_EX, il existe quatre familles de fontes : 'Sérf', 'Sans Sérf', 'Télétype' ou 'Math'. Notez que chaque jeu de fontes se spécialise souvent dans une famille de fonte. Toutefois, quelques-unes fournissent toutes les familles d'un coup.
- La **série** : Correspond à l'espacement entre les caractères et l'épaisseur du trait. Bien qu'il en existe cinq types de séries au niveau de T_EX (*medium*, *bold*, *bold extended*, *semi-bold*, *condensed*), seuls deux sont directement accessibles au niveau de L^AT_EX : Moyen (*medium*) et Gras (*bold*).
- La **forme** : Correspond à l'inclinaison des caractères ou encore des caractéristiques annexes qui permettent de faire ressortir le mot ou la phrase du texte qui l'environne.
- La **taille** : Correspond à la taille des caractères.

La table 4 liste les principaux jeux de fontes de L^AT_EX⁷. La table 5 liste les familles, les séries et les formes de fontes qui existent. Et, enfin, la table 6 liste les différentes tailles de fontes accessibles.

Enfin, la *décoration* est un attribut, non obligatoire, que l'on peut ajouter à un texte. Il s'agit d'une caractéristique supplémentaire qui vient s'ajouter en fonction des paramètres déjà existants de la fonte. Par exemple, l'*emphase* est une décoration qui sert à faire ressortir le texte choisi. Il joue sur l'orientation des caractères (oblique ou droit) en prenant l'inverse du contexte dans lequel il se trouve. Ainsi, si on se trouve dans un paragraphe mis en italique, l'emphase aura pour effet de donner des caractères droits, et inversement. La table 7 récapitule quelques décorations. Notez qu'il en existe bien plus en mode mathématiques que dans le mode normal, mais nous le verrons par la suite.

Questions

1. Reprenez votre fichier `mydocument.tex` et essayez quelques fontes proposées dans la table 4.
2. Produisez le rendu suivant : A B C D E F G H I J
3. Trouvez ce qui donne le code suivante : `\emph{Ceci est un texte en \emph{emphase}.}`

<input type="checkbox"/> Ceci est un texte en <i>emphase</i> .	<input type="checkbox"/> Ceci est un texte en <i>emphase</i> .
<input type="checkbox"/> <i>Ceci est un texte en emphase.</i>	<input type="checkbox"/> <i>Ceci est un texte en emphase.</i>

5. Les fontes par défaut sont souvent suffisantes, mais le jeu de fontes 'Latin Modern' (`lmodern`) est plus complet.

6. Celui-ci est chargé dans votre document via la commande : `\usepackage[T1]{fontenc}`

7. Pour une liste plus exhaustive des jeux de fontes, voir : <http://www.tug.dk/FontCatalogue/>

Nom	Famille(s)	Commande
Computer Modern	Toutes	fontes par défaut
ASCII*	Télétype	<code>\usepackage{ascii}</code>
Avantgarde*	Sans Sérif	<code>\usepackage{avant}</code>
Bookman*	Sérif	<code>\usepackage{bookman}</code>
Charter Bistream	Sérif/Math	<code>\usepackage[charter-bistream]{mathdesign}</code>
Courier*	Télétype	<code>\usepackage{courier}</code>
DejaVu	Toutes	<code>\usepackage{dejavu}</code>
Euler	Math	<code>\usepackage{euler}</code>
Fourier Utopia	Sérif/Math	<code>\usepackage{fourier}</code>
Fourier New Century	Sérif/Math	<code>\usepackage{fouriernc}</code>
Helvetica*	Sans Sérif	<code>\usepackage{helvet}</code>
Iwona	Sans Sérif/Math	<code>\usepackage[math]{iwona}</code>
Latin Modern	Toutes	<code>\usepackage{lmodern}</code>
Palatino	Sérif/Math	<code>\usepackage{mathpazo}</code>
Times	Sérif/Math	<code>\usepackage{mathptmx}</code>
Zapf Chancery*	Sérif	<code>\usepackage{chancery}</code>

TABLE 4 – Les différents jeux de fontes (* marque les jeux de fontes sans mode mathématique).

Usage		Rendu	Commande	
			Argument	Contexte
Familles	Sérif	AaBbCcDdEeFf	<code>\textrm{...}</code>	<code>{\rmfamily ...}</code>
	Sans Sérif	AaBbCcDdEeFf	<code>\textsf{...}</code>	<code>{\sffamily ...}</code>
	Télétype	AaBbCcDdEeFf	<code>\texttt{...}</code>	<code>{\ttfamily ...}</code>
	Math	<i>AaBbCcDdEeFf</i>	–	<code>\(...\)</code> ou <code>\$...\$</code>
Séries	Moyen	AaBbCcDdEeFf	<code>\textmd{...}</code>	<code>{\mdseries ...}</code>
	Gras	AaBbCcDdEeFf	<code>\textbf{...}</code>	<code>{\bfseries ...}</code>
Formes	Droite	AaBbCcDdEeFf	<code>\textup{...}</code>	<code>{\upshape ...}</code>
	Oblique	<i>AaBbCcDdEeFf</i>	<code>\textsl{...}</code>	<code>{\slshape ...}</code>
	Petites capitales	AaBbCcDdEeFf	<code>\textsc{...}</code>	<code>{\scshape ...}</code>
	Italique	<i>AaBbCcDdEeFf</i>	<code>\textit{...}</code>	<code>{\itshape ...}</code>

TABLE 5 – Les différents types de fontes en L^AT_EX.

Rendu	Commande
AaBbCcDdEeFf	<code>{\Huge ...}</code>
AaBbCcDdEeFf	<code>{\huge ...}</code>
AaBbCcDdEeFf	<code>{\LARGE ...}</code>
AaBbCcDdEeFf	<code>{\Large ...}</code>
AaBbCcDdEeFf	<code>{\large ...}</code>
AaBbCcDdEeFf	<code>{\normalsize ...}</code>
AaBbCcDdEeFf	<code>{\small ...}</code>
AaBbCcDdEeFf	<code>{\footnotesize ...}</code>
AaBbCcDdEeFf	<code>{\scriptsize ...}</code>
AaBbCcDdEeFf	<code>{\tiny ...}</code>

TABLE 6 – Les différentes tailles de fontes disponibles.

Usage	Rendu	Commande
Normal	AaBbCcDdEeFf	<code>\textnormal{...}</code> ou <code>{\normalfont ...}</code>
Emphase	<i>AaBbCcDdEeFf</i>	<code>\emph{...}</code> ou <code>{\emph ...}</code>
Souligné	<u>AaBbCcDdEeFf</u>	<code>\underline{...}</code> ou <code>{\underline ...}</code>

TABLE 7 – Quelques types de décorations en L^AT_EX.

4. Mettre en relation les phrases (à gauche) et les caractéristiques (famille, série, forme et taille) à droite :

- | | |
|--|----------------------------------|
| a. Portez ce vieux whisky au juge blond qui fume. | 1. Normale. |
| b. <i>Portez ce vieux whisky au juge blond qui fume.</i> | 2. Sans-sérif, Large. |
| c. <small>PORTEZ CE VIEUX WHISKY AU JUGE BLOND QUI FUME.</small> | 3. Télétipe, Petites capitales. |
| d. Portez ce vieux whisky au juge blond qui fume. | 4. Petite capitales, Scriptsize. |
| e. <small>PORTEZ CE VIEUX WHISKY AU JUGE BLOND QUI FUME.</small> | 5. Oblique, Gras, Small. |

2.2 Structure d'un document

La notion de structure d'un document est extrêmement importante en L^AT_EX (et dans l'activité d'écriture en général). C'est pour cela que beaucoup de commandes aident à accomplir cette tâche plus facilement. Notamment, il est possible de structurer très finement son document en le découpant en partie hiérarchiques. La table 8 liste toutes les commandes permettant de structurer un document.

Usage	Niveau	Partition
Déclare le début d'une partie (n'existe que dans <code>book</code>)	-1	<code>\part{title}</code>
Déclare le début d'un chapitre (n'existe pas dans <code>article</code>)	0	<code>\chapter{title}</code>
Déclare le début d'une section	1	<code>\section{title}</code>
Déclare le début d'une sous-section	2	<code>\subsection{title}</code>
Déclare le début d'une sous-sous-section	3	<code>\subsubsection{title}</code>
Déclare le début d'un paragraphe	4	<code>\paragraph{title}</code>
Déclare le début d'un sous-paragraphe	5	<code>\subparagraph{title}</code>
Déclare une partition <code><name></code> mais sans numérotation	–	<code>\<name>{*}{title}</code>
Marque le début des appendices	–	<code>\appendix</code>

TABLE 8 – Les différentes partitions possibles d'un document.

Pour compléter cette notion de structure du document, il est possible de créer une page de titre ainsi que diverses tables (table des matières, table des figures, table des tables, ...).

Usage	Commande
Positionne le titre du document	<code>\title{mytitle}</code>
Positionne le nom des auteurs	<code>\author{myname}</code>
Positionne la date (optionnel)	<code>\date{today}</code>
Fait afficher le titre	<code>\maketitle</code>
Fait afficher la table des matières	<code>\tableofcontents</code>
Fait afficher la table des figures	<code>\listoffigures</code>
Fait afficher la table des tables	<code>\listoftables</code>

TABLE 9 – Définition de la page de titre et les différentes tables.

Questions

1. Reprenez votre document d'exemple, ajoutez lui un titre et un auteur. Puis, produisez le document final successivement pour la classe `article`, `report`, `book` et `memoir`.
2. Toujours sur l'exemple, créez la hiérarchie affichée sur le listing 2 et faites afficher la table des matières (à nouveau, testez le rendu suivant les classes de documents : `article`, `report`, `book` et `memoir`).
3. Ajoutez un chapitre d'appendice à votre document, comme le suggère le listing 3. Puis, visualisez le rendu suivant les différentes classes de document.

```
\maketitle

\begin{abstract}
Ceci est un résumé du document.
\end{abstract}

\tableofcontents

\chapter*{Introduction}
\section{Une première section}
Ceci est un texte de section.

\section{Une second section}
\subsection{Une sous-section}

\part{Ma première partie}
\chapter{Mon premier chapitre}
\section{Une première section}
\section{Une second section}
\subsection{Une sous-section}
```

Listing 2 – Un exemple de structure de document L^AT_EX.

```
\appendix

\chapter{Un chapitre en appendice}
\section{Une première section}
\section{Une second section}
\subsection{Une sous-section}
```

Listing 3 – Un exemple d'appendix en L^AT_EX.

2.3 Paragraphes

La notion de paragraphe en L^AT_EX est obtenue en insérant (au moins) une ligne vide entre deux blocs de textes. Par exemple, le texte suivant est composé de deux paragraphes :

```
Lorem ipsum dolor sit amet,
consectetur adipiscing
elit.

Praesent vel eros leo,
sollicitudin laoreet mi.
```

Listing 4 – Texte de deux paragraphes.

Il existe un certain nombre de commandes pour forcer le formatage des paragraphes localement (voir table 10). Ainsi, on peut jouer sur l'ajustement, l'indentation de début de paragraphe, l'espace inter-paragraphes, *etc.* Tout ceci dans le but d'améliorer le rendu et le confort de lecture.

```

Lorem ipsum dolor sit amet,
consectetur adipiscing
elit.
Praesent vel eros leo,
sollicitudin laoreet mi.

```

Listing 5 – Texte d’un seul paragraphe.

Attention: Il est possible de changer globalement les paramètres pour l’ensemble du document. Ce sera parfois ce dont vous aurez besoin, mais nous n’aborderons pas ce genre de choses ici.

Usage	Commande
Ajusté à gauche (env, context)	<code>flushleft</code> ou <code>\raggedright</code>
Ajusté à droite (env, context)	<code>flushright</code> ou <code>\raggedleft</code>
Ajusté centré (env, context)	<code>center</code> ou <code>\centering</code>
Force l’indentation initiale	<code>\indent</code>
Supprime l’indentation initiale	<code>\noindent</code>
Force le passage à la ligne	<code>\newline</code> ou <code>\\</code>
Passage à la ligne ajustable	<code>\\[1em]</code>
Force le passage à la page	<code>\newpage</code>
Petit saut inter-paragraphe	<code>\smallskip</code>
Moyen saut inter-paragraphe	<code>\medskip</code>
Grand saut inter-paragraphe	<code>\bigskip</code>
Saut vertical ajustable	<code>\vspace{1em}</code> ou <code>\\[1em]</code>
Saut horizontal ajustable	<code>\hspace{1em}</code>
Rempli l’espace vertical	<code>\vfill</code>
Rempli l’espace horizontal	<code>\hfill</code>

TABLE 10 – Les commandes pour formater les paragraphes.

Questions

1. Faites un paragraphe sans indentation initiale, suivi d’un paragraphe normal.
2. Faites un paragraphe justifié à gauche, une autre centré, et, enfin, un paragraphe justifié à droite.
3. Espacez deux paragraphes consécutifs de 4em.
4. Essayez de reproduire le texte suivant :
Ceci est un essai de texte qui s’étale sur toute la ligne.

2.4 Listes

Les listes permettent d’énumérer de façon lisible un certain nombre d’éléments. Ces éléments peuvent être ordonnés ou non. En \LaTeX , la syntaxe générale d’une liste est présentée dans le listing 6. Et la table 11 résume les principales commandes qui permettent de faire des listes.

Questions

1. Faites une liste non-ordonnée, une liste ordonnée et une liste de descriptions.
2. Reproduisez la liste suivante :
— First level, itemize, first item

```

\begin{itemize}
\item Ceci est le premier point de la liste.
\item[Deux] Ceci est le deuxième point de la liste.
\item Ceci est le point final de la liste.
\end{itemize}

\begin{description}
\item[Biology] Study of life.
\item[Physics] Science of matter and its motion.
\item[Psychology] Scientific study of mental processes and behaviour.
\end{description}

```

Listing 6 – Exemples de listes d’éléments.

Usage	Commande
Liste non-ordonnée	<code>itemize</code>
Liste ordonnée	<code>enumerate</code>
Liste de descriptions	<code>description</code>
Élément de la liste	<code>\item</code>
Élément nommé de la liste	<code>\item[a]</code>

TABLE 11 – Les commandes pour créer des listes.

- Second level, itemize, first item
- Second level, itemize, second item
 1. Third level, enumerate, first item
 2. Third level, enumerate, second item
- First level, itemize, second item

2.5 Tableaux

Comme pour tout le reste, les tableaux sont aussi décrits de manière textuelle. On décrit les tableaux sous la forme d’un environnement où la première ligne donne toutes les spécifications d’une des lignes du tableau comme ceci : `\begin{tabular}[vpos]{linespec}`

L’argument ‘`linespec`’ permet de définir combien de cellules seront définies par ligne, leur alignement horizontal respectif et la présence, ou pas, de ligne de séparation entre elles. De plus, l’option ‘`vpos`’ permet de contrôler l’alignement vertical du contenu des cellules. Le listing 7 présente un exemple de tableau spécifié en \LaTeX . La table 12 présente les différentes commandes qui permettent de définir un tableau.

```

\begin{tabular}[m]{|c||l|r}
    centrée & alignée à gauche & alignée à droite \\
\cline{2-3}
    Une autre & ligne dans & le tableau \\
\hline
\end{tabular}

```

Listing 7 – Exemple de tableau simple.

Évidemment, il ne s’agit là que de tableaux ‘simples’. Il est néanmoins possible de réaliser des tableaux plus complexes grâce aux commandes `\multicol` (qui fait partie des commandes standards) et `\multirow`

Usage	Commande
Position optionnelle	
Aligné sur le bas	b
Centrée verticalement (défaut)	c
Aligné sur le haut	t
Spécification des lignes	
Cellule alignée à gauche	l
Cellule centrée	c
Cellule alignée à droite	r
Barre verticale	
Double barre verticale	
Séparateurs utilisés dans le tableau	
Début d'une nouvelle cellule	&
Début d'une nouvelle ligne	\\
Ligne de séparation horizontale	\hline
Ligne de séparation partielle	\cline{2-4}

TABLE 12 – Les différents attributs de l'environnement tableau.

(qui fait parti du paquet du même nom⁸). Ces deux commandes permettent de grouper plusieurs cellules entre-elles pour n'en former qu'une, soit verticalement, soit horizontalement. Un exemple utilisant les deux commandes est donné dans le listing 8, le rendu résultant est montré par la table qui est à côté.

La commande `\multicolumn{nbcols}{format}{cellcontent}` a trois arguments :

- `nbcols` : Le nombre de colonnes sur lequel vont s'étendre la ou les cellules.
- `format` : Le format interne de la cellule.
- `cellcontent` : Le contenu des cellules.

La commande `\multirow{nbrows}{width}{cellcontent}` a aussi trois arguments :

- `nbrows` : Le nombre de lignes sur lequel va s'étendre la cellule.
- `width` : La largeur de cette cellule ('*' signifie qu'elle prend automatiquement l'espace nécessaire).
- `cellcontent` : Le contenu de la cellule.

```
\begin{tabular}{|l|l|l|l|}
\hline
\multicolumn{3}{|c|}{Team sheet} \\
\hline
Goalkeeper & GK & Paul Robinson \\
\multirow{4}{*}{Defenders} & LB & Lucas Radebe \\
& DC & Michael Duberry \\
& DC & Dominic Matteo \\
& RB & Didier Domi \\
\multirow{3}{*}{Midfielders} & MC & David Batty \\
& MC & Eirik Bakke \\
& MC & Jody Morris \\
Forward & FW & Jamie McMaster \\
\multirow{2}{*}{Strikers} & ST & Alan Smith \\
& ST & Mark Viduka \\
\hline
\end{tabular}
```

Team sheet		
Goalkeeper	GK	Paul Robinson
Defenders	LB	Lucas Radebe
	DC	Michael Duberry
	DC	Dominic Matteo
	RB	Didier Domi
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Forward	FW	Jamie McMaster
Strikers	ST	Alan Smith
	ST	Mark Viduka

Listing 8 – Exemple de tableau complexe.

8. Pour le charger faites : `\usepackage{multirow}`

Questions

1. Reproduisez les tableaux suivants :

1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
7	8	9	7	8	9	7	8	9	7	8	9	7	8	9

2. Reproduisez les tableaux suivants :

7C0		hexadecimal
3700		octal
11111000000		binary
1984		decimal

		Primes			
		2	3	5	7
Powers	504	3	2	0	1
	540	2	3	1	0
Powers	gcd	2	2	0	0
	lcm	3	3	1	1

min
max

2.6 Images

L'insertion d'images dans le texte se fait grâce au paquet `graphicx` qu'il vous faudra charger au début de votre fichier `LATEX`. Une fois ceci fait, charger une image dans votre texte se fait simplement via la commande suivante (voir table 13 pour les options) : `\includegraphics[opt1=val1,...,optn=valn]{imagename}`

Usage	Attribut
Largeur de l'image	<code>width=xx</code>
Hauteur de l'image	<code>height=xx</code>
Zoom de l'image	<code>scale=xx</code>
Rotation de l'image	<code>angle=xx</code>
Passe en mode ' <i>clipping</i> '	<code>clip</code>
Partie de l'image à couper	<code>trim=l b r t</code>

TABLE 13 – Les différents attributs de `includegraphics`.**Questions**

- Prenez une image quelconque et reproduisez les visuels obtenus sur les figures 2 et 3.
- En cherchant dans la documentation, essayez de faire un clipping de l'image que vous utilisez.

2.7 Références et notes de bas de page

`LATEX` propose un système pour gérer de façon dynamique, toutes les références croisées, les notes en bas de pages ainsi que les notes dans la marge du document. Cela évite ainsi, après avoir bougé un bloc de votre texte, d'avoir à tout renuméroter. Il suffit de laisser la machine faire cela pour vous.

Usage	Commande
Poser un label	<code>\label{...}</code>
Faire une référence	<code>\ref{...}</code>
Page de la référence	<code>\pageref{...}</code>
Note de bas de page	<code>\footnote{...}</code>
Note sur la marge droite	<code>\marginpar{...}</code>
Inverse la marge utilisée	<code>\reversemarginpar</code>

Usage	Label
Chapitre	<code>chap:</code>
Section	<code>sec:</code>
Figure	<code>fig:</code>
Table	<code>tab:</code>
Équation	<code>eq:</code>
Listing	<code>lst:</code>

TABLE 14 – Les commandes relatives aux références croisées.

La table 14 (à gauche) liste toutes les commandes relatives à ces références. Même si ces commandes sont relativement explicites, il faut préciser quelques usages habituels que les gens avertis font de ces commandes :



FIGURE 2 – Exemple d'image insérée dans le document (`width=\linewidth`).



FIGURE 3 – Exemple d'images (suite) : réduite (`width=.35\linewidth`) et tournée (`angle=45`).

- Les références croisées se font en accolant à une section, chapitre, figure, table, *etc*, la commande `\label{...}`. Il est aussi habituel de préfixer le label qu'on lui accole par un identifiant (voir table 14 (à droite)) afin de bien identifier l'objet qui est référencé. Cela peut souvent éviter des confusions dommageable pour le document.
- La commande `\ref` s'utilise toujours en ajoutant une espace insécable entre le mot référencé et la référence (le '~' code une espace insécable en \LaTeX) : `mot~\ref{...}`.
- Les notes de bas de page, se placent accolé au mot référencé : `mot\footnote{Ceci est ma note.}`
- Enfin, les notes dans les marges sont uniquement utilisées sur les brouillons pour indiquer des parties de votre document à retravailler (TODO, FIXME) ou des notes à l'intention des autres auteurs.

Questions

1. Expérimentez l'usage d'une référence croisée sur une section, puis une note de bas de page et, enfin, une note en marge.
2. Essayez de faire une note de bas page depuis un mot à l'intérieur d'un tableau.

2.8 Objets flottants

Dans toute composition de document, il existe un certain nombre d'objets typographiques qui sont insécables (figures, tableaux, ...). Ceux-ci posent problème lorsque leur positionnement n'est pas possible dans une page et qu'ils doivent apparaître sur la page suivante. Ce qui laisse, inévitablement, un grand vide dans la page où ils auraient dû apparaître.

C'est en grande partie pour éviter ce genre de désagrément que les typographes ont inventé la notion d'*objets flottants* (ou juste *flottants*). Ces objets, bien que relié au texte, peuvent se placer approximativement là où ils ont été décrit mais laissent une certaine marge de manœuvre au compilateur.

Laisser \LaTeX gérer le positionnement de ces objets vous facilitera la vie pendant la phase de rédaction du document et vous allégera considérablement la phase de mise en page finale. C'est donc un bon pari à prendre si vous voulez vous éviter du travail (surtout dans le rush final de la mise en page).

Il existe un grand nombre d'objets flottants mais nous nous focaliserons ici seulement sur les figures (voir listing 9) et les tableaux (voir listing 10).

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=\linewidth]{Tigers}
  \caption{Ceci est un exemple de figure.}
  \label{fig:figure-example}
\end{figure}
```

Listing 9 – Figure flottante avec une légende.

```
\begin{table}[htbp]
  \centering
  \begin{tabular}{lcr}
    1 & 2 & 3 \\
    4 & 5 & 6 \\
    7 & 8 & 9 \\
  \end{tabular}
  \caption{Ceci est un exemple de table.}
  \label{tab:table-example}
\end{table}
```

Listing 10 – Table flottante avec une légende.

Même si le placement de ces objets est laissé au compilateur, il est possible de passer des indications au compilateur pour faire pencher la balance plutôt du côté de votre choix. La table 15 résume les différentes options qui permettent de passer ces options au compilateur.

Usage	Option
Ici (<i>here</i>)	<code>h</code>
Haut de page (<i>top</i>)	<code>t</code>
Bas de page (<i>bottom</i>)	<code>b</code>
Dans la page (<i>page</i>)	<code>p</code>
Force le placement "Ici"	<code>h!</code>
Placements essayés dans l'ordre	<code>htbp</code>

TABLE 15 – Les différentes options de placement des flottants.

Questions

1. Faites une figure et un tableau, puis essayez de les déplacer à divers endroits de votre document \LaTeX .
2. En plaçant la figure à un endroit où elle saute à la page suivante, essayez les différentes options de positionnement et observez leurs effets.

3 Composition avancée de documents \LaTeX

Les notions de base du \LaTeX étant assimilées, nous allons voir quelques modules spécialisés pour les traitements un peu plus spécifiques que d'écrire du texte. Nous allons aborder le mode mathématique, la gestion d'une bibliographie avec \BibTeX , la réalisation de schémas avec \TikZ et la réalisation de diapos pour une présentation orale.

3.1 Mathématiques

Le mode mathématiques de \TeX est la principale raison d'utiliser \LaTeX plutôt qu'autre chose. D'abord parce qu'il permet un rendu typographique qui reste, encore aujourd'hui, inégalé parmi les autres outils de formatage de texte. Ensuite, parce qu'après une petite période d'apprentissage nécessaire, il devient une vraie seconde nature. Ainsi, il n'est pas rare de voir les utilisateurs de \LaTeX écrire des équations directement dans le format texte du langage lorsqu'ils communiquent par e-mails sans moteur de rendu derrière.

Quoiqu'il en soit, même si le mode mathématiques est toujours chargé par défaut, certains petites extensions peuvent parfois manquer. Si vous vous préparez à utiliser un tant soit peu le mode mathématique, il est toujours bon de charger le paquet '`mathtools`' (avec `\usepackage{mathtools}`).

3.1.1 Les modes mathématiques

\LaTeX propose trois types de mode mathématiques, l'un permet d'insérer des formules mathématiques dans un flot de texte (*inline*) et le second permet d'afficher les formules en les mettant en valeur séparée du texte (*display*) et le dernier est une variation de '*display*' avec une numérotation de la formule (*numbered*) qui permet d'y faire référence. La table 16 montre les différentes syntaxes qui permettent d'appeler ces différents modes. Notez qu'il est toujours préférable d'utiliser la syntaxe \LaTeX pour diverses raisons de compatibilité et de lisibilité de votre code.

Type	Environnement	\LaTeX	\TeX
Inline	<code>\begin{math} ... \end{math}</code>	<code>\(... \)</code>	<code>\$... \$</code>
Display	<code>\begin{displaymath} ... \end{displaymath}</code>	<code>\[... \]</code>	<code>\$\$... \$\$</code>
Numbered	<code>\begin{equation} ... \end{equation}</code>	—	—

TABLE 16 – Les différents types et syntaxes du mode `math`.

En fait, il existe déjà beaucoup de documentations sur l'apprentissage du mode `math` de \LaTeX , reportez-vous à l'une de ces documentations pour en savoir plus⁹ et pour faire les exercices qui suivent.

Questions

1. Reproduisez les formules suivantes (conservez l'alignement des formules 1 et 2 entre elles et munissez les équations numérotées de labels permettant un appel via un `\ref`) :

$$\Delta t' = \frac{\Delta t}{\sqrt{1 - \frac{v^2}{c^2}}} \quad E = \gamma mc^2 \quad (1) \quad \Delta K = \int_{t_0}^{t_1} \frac{d}{dt} (\gamma m \mathbf{v}) \cdot \mathbf{v} dt$$

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (2) \quad (3)$$

9. Voir : <http://en.wikibooks.org/wiki/LaTeX/Mathematics>, http://en.wikibooks.org/wiki/LaTeX/Advanced_Mathematics.

2. Reproduisez les matrices suivantes :

$$A_{m,n} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix} \qquad \begin{pmatrix} \varphi_1 & & 0 \\ & \varphi_2 & \\ 0 & & \ddots \\ & & & \varphi_n \end{pmatrix}$$

3. Reproduisez le théorème 1 et sa démonstration :

Théorème 1. *Il existe des nombres irrationnels a et b , tels que a^b est un nombre rationnel.*

Démonstration. On sait que 2 est rationnel et $\sqrt{2}$ irrationnel. On suppose que $\sqrt{2}^{\sqrt{2}}$ est rationnel :

- Si notre supposition initiale est vraie, alors $a = \sqrt{2}$ et $b = \sqrt{2}$ conviennent.
- Si notre supposition initiale est fausse, alors $\sqrt{2}^{\sqrt{2}}$ est irrationnel et, donc, $a = \sqrt{2}^{\sqrt{2}}$ et $b = \sqrt{2}$ conviennent puisque $a^b = 2$.

Quelle que soit la validité de l'hypothèse initiale, le théorème 1 est vrai, ce qui prouve sa validité. \square

3.2 Bibliographie

Dans tout document scientifique, l'importance des références est cruciale. La pertinence de vos sources et la façon dont vous les citez peut selon le sérieux avec lequel vous le faites, vous apporter un certain crédit aux yeux du lecteur ou vous discréditer totalement. L^AT_EX propose un système très complet pour citer vos sources simplement à l'intérieur d'un document. Celui-ci s'appuie sur un mécanisme comparable aux références croisées via la commande `\cite` (voir table 17 à gauche).

De plus, l'outil BibT_EX permet de gérer facilement toutes vos références et de constituer une base bibliographique que vous pourrez réutiliser d'un document sur l'autre en la combinant avec les commandes L^AT_EX déjà mentionnées. Nous allons voir ici comment tout cela fonctionne.

3.2.1 Principes de BibT_EX

BibT_EX permet de constituer une liste de références dans un fichier texte tout simple. Le format de ce fichier est imposé par BibT_EX et possède toujours une extension en `‘.bib’`. Le principe est que l'on charge ce fichier dans le document L^AT_EX via la commande `\bibliography{...}` (et, éventuellement, que l'on choisisse son style de bibliographie avec `\bibliographystyle{...}`).

Une fois ceci fait, chaque citation appelée via une commande `\cite{...}` apparaîtra dans la liste de vos références avec le style que vous aurez choisi. Cela permet, notamment, de constituer une large base de citations qui peut ensuite être utilisée par plusieurs documents sans avoir à l'inclure dans son intégralité.

Notez aussi que, tout comme pour les références croisées, il est d'usage d'ajouter une espace insécable `‘~’` entre le mot auprès duquel on met la citation et la citation : `mot~\cite{...}`.

Néanmoins, pour obtenir le rendu final du document, il vous faudra rajouter une étape supplémentaire dans la compilation en faisant appel à l'outil `bibtex`. Les quatre étapes sont les suivantes :

1. `pdflatex mydocument`¹⁰ : Crée les fichiers auxiliaires `‘.aux’` qui serviront de support pour déterminer quelles références sont appelées via les commandes `\cite{...}` ou `\nocite{...}`.
2. `bibtex mydocument` : Crée le fichier `.bbl` qui correspond au code T_EX qui sera inséré dans le rendu final (crée aussi un fichier `.blg` qui renferme un log de la compilation).
3. `pdflatex mydocument` : Détermine le nombre de citations et repère les endroits où elles seront insérées.
4. `pdflatex mydocument` : Raffine les références des citations et insère les références dans le texte final.

¹⁰. Notez que l'on a pas mis l'extension du fichier `mydocument.tex`. Cette syntaxe évite certaines ambiguïtés et elle est particulièrement recommandée. Surtout lorsque vous faites usage de BibT_EX.

Usage	Commande	Style	Nom
Cite une référence	<code>\cite{HG75}</code>	Style abrégé	abbrv
Cite plusieurs références	<code>\cite{Book1,Article2}</code>	Style ACM	acm
Note après la citation : [BK98, p. 34]	<code>\cite[p.~34]{BK98}</code>	Style alphabétique	alpha
Force l’affichage de la référence	<code>\nocite{Article1}</code>	Style APA	apalike
Force l’affichage de toutes les références	<code>\nocite{*}</code>	Style IEEE	ieeetr
Positionne le style bibliographique	<code>\bibliographystyle{...}</code>	Style standard	plain
Affiche les références de <code>mybiblio.bib</code>	<code>\bibliography{mybiblio}</code>	Style SIAM	siam
		Style non-trié	unsrt

TABLE 17 – Les commandes pour gérer la bibliographie en L^AT_EX et quelques styles classiques.

```
...
\nocite{*}
\bibliographystyle{plain}
\bibliography{bibliography}
...
```

Listing 11 – Bibliographie en L^AT_EX.

```
@Book{GMS93,
  author   = "Goossens, Michel and
             Mittelbach, Franck and
             Samarin, Alexander",
  title    = "{The {\LaTeX} Companion}",
  year     = 1993,
  publisher= "Addison-Wesley",
  address  = "Reading, Massachusetts"
}

@Misc{website:google,
  author   = "Google Inc.",
  title    = "Google Web Search Engine",
  month    = "August",
  year     = 2012,
  howpublished= "\url{http://www.google.com/}"
}
```

Listing 12 – Contenu de `bibliography.bib`.

3.2.2 Format BibT_EX

Le format de BibT_EX consiste en un ensemble de catégories de documents que l’on référence (voir listing 12] et table 18). Chacune de ces catégorie rassemble ensuite un label (en première position), puis un ensemble de champs, dont certains sont obligatoires. Notez que si vous utilisez `emacs`, cet éditeur possède un mode spécial pour les fichiers BibT_EX, il suffit d’aller dans le menu ‘*Entry-Types*’ pour avoir accès à une liste de templates d’entrées qu’il ne reste plus qu’à remplir.

Usage	Type d’entrée
Article dans un journal ou magazine	@Article
Article dans une conférence	@Inproceedings
Documentation technique	@Manual
Document non encore publié	@Unpublished
Livre avec un éditeur	@Book
Livre sans éditeur	@Booklet
Partie, chapitre ou section d’un livre	@Inbook
Mémoire de Master	@Mastersthesis
Mémoire de Thèse	@Phdthesis
Autres (site web, logiciel, ...)	@Misc

TABLE 18 – Les différents types d’entrées de BibT_EX.

Il faut savoir que l’on peut insérer du L^AT_EX dans les champs en question, mais certains champs perdent la distinction majuscule/minuscule (comme le champs du titre, par exemple). Pour forcer BibT_EX à conserver le titre tel quel, il faut le protéger avec des accolades ‘{...}'. Enfin, le champs des auteurs est très particulier car il a une syntaxe qui lui est propre. Les règles sont les suivantes :

- Chaque auteur est séparé des autres par un ‘and’.
- On inscrit le nom de chaque auteur ainsi : `nom, prénom_2 prénom_2`
- Mettez les prénoms en entier car le style `abbrv` permet de n’en citer que les initiales automatiquement.
- Les particules d’un nom doivent être toujours mises en minuscule : ‘von Neumann, John’
- Si l’on a un suffixe, la syntaxe devient : `nom, suffixe, prénom`.
Par exemple : ‘Bush, Jr., George Walker’ et ‘Williams, III, Hanks’.

Questions

1. Ajoutez une bibliographie dans votre document de travail, compilez la et essayez différents styles.
2. Trouvez les références de l’article ‘*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*’, créez une entrée adéquate dans votre fichier bibliographique et citez le dans votre document.

3.3 Créer des schémas avec TikZ

Il arrive souvent qu’un schéma ou une figure soit bien plus explicite et agréable que des pages et des pages d’explications. Mais, les images ont aussi leurs inconvénients. Tout d’abord, elles sont beaucoup plus longues à réaliser. Et, souvent, vous aurez recours à des outils extérieurs à \LaTeX pour les créer. Mais, dans ce cas, vous vous rendrez compte que ce que vous obtiendrez sera d’une qualité largement inférieure au rendu global du document, ou n’utilisera pas la même fonte que le reste du document, ou encore ne sera pas à la bonne taille pour se couler dans la mise en page. Quoiqu’il en soit, très souvent votre image tranchera par rapport à l’ensemble de votre document. Qui plus est, ces images augmenteront considérablement le poids du document final en mémoire. Certains documents peuvent même parfois atteindre des tailles qui rendent difficile, voire impossible, un envoi par courrier électronique. Ce qui est assez handicapant lorsqu’on veut rendre un rapport.

Afin de répondre à ces problématiques, plusieurs langages de descriptions de schémas existent en \LaTeX , mais, de nos jours, le plus fréquemment utilisé est sans conteste TikZ. Grâce à une grande flexibilité de son langage, une grande puissance d’expression et un système de paquetages qui permet une certaine modularité, TikZ est un langage de description graphique qui a fait la différence dans le monde \LaTeX . Il reste, néanmoins, un langage difficile à approcher lorsqu’on débute et requiert une certaine pratique avant d’en comprendre les bases et encore plus de travail pour en comprendre les finesses. Mais le bénéfice est important car les schémas obtenus sont vraiment d’une qualité quasi-inégalée et avec une intégration de \LaTeX qui est quasiment parfaite. De plus, la grosseur des images est minimale car elles sont décrites sous forme vectorielle¹¹. Pour donner une idée des possibilités de TikZ, reportez vous à la figure 4 qui présente un exemple de figure faites avec TikZ. Comme vous le voyez, les figures TikZ peuvent être aussi complexes que vous le désirez, la seule limite est votre capacité à maîtriser le langage.

Évidemment, le but de cette feuille d’exercices n’est pas de vous faire découvrir les finesses de TikZ, nous ne ferons qu’effleurer le sujet et nous n’aborderons que les bases du langage. Il est donc conseillé de vous reporter à la section 3.3.4 qui liste des références plus exhaustives qui vous permettront de vous perfectionner dans ce langage. Aussi, gardez à l’esprit que la plupart des gens ne partent quasiment jamais d’une feuille blanche. Il est fréquent de trouver sur Internet un exemple de schéma similaire à ce que l’on veut réaliser et de partir de là. N’hésitez donc pas à pratiquer cette technique, c’est aussi une bonne façon d’apprendre en se confrontant au code des autres.

3.3.1 Paquets, bibliothèques et environnements pour TikZ

Utiliser TikZ requiert une configuration minimale. Tout d’abord, il faut charger le paquet \LaTeX ‘`\usepackage{tikz}`’ avant le ‘`\begin{document}`’.

Ensuite, comme nous l’avons évoqué plus haut, TikZ a son propre système de bibliothèques et ceux-ci sont chargés par la commande ‘`\usetikzlibrary{liste des bibliothèques séparées par des virgules}`’. Les bibliothèques sont très nombreuses (et variées), par exemple on peut citer : `arrows`, `automata`, `backgrounds`, `calendar`, `chains`, `matrix`, `mindmap`, `patterns`, `petri`, `shadows`, `shapes.misc`, `shapes.geometric`, `spy`, `trees`, ... et d’autres encore.

11. C’est à dire comme la somme de primitives traçant les points, les lignes, les cercles, *etc*, qui composent le schéma et non comme un ensemble de points. Au passage, agrandir une image vectorielle se fait sans perte de précision alors que ce n’est pas le cas pour une image “classique”.

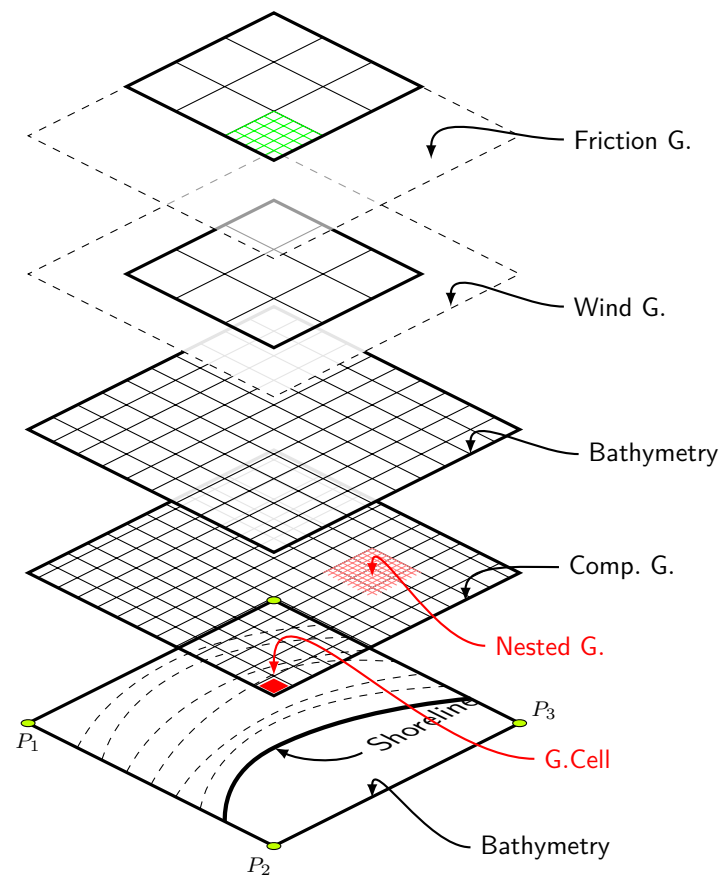


FIGURE 4 – Exemple de figure complexe en $\text{\textit{TikZ}}$ trouvée sur <http://texample.net> (par Marco MIANI).

Enfin, il existe deux façon d'intégrer les schémas TikZ dans un document. Soit en les intégrant dans le texte via la commande :

```
\tikz[<options>]{<tikz commands>}
```

Soit, en les encapsulant dans un environnement, par exemple pour les présenter au sein d'une figure :

```
\begin{tikzpicture}[<options>]
  <tikz commands>
\end{tikzpicture}
```

3.3.2 Principes du langage TikZ

La syntaxe du langage emprunte beaucoup à L^AT_EX, on y retrouve la syntaxe d'une commande et la façon de passer des options de manière classique. Il existe néanmoins une différence de taille qui est que les commandes doivent toutes se terminer par un point-virgule, sans cela un message d'erreur sera lancé :

```
\command[<options>]<specification>;
```

Bien que l'on puisse croire le contraire, il n'existe en fait que deux types de commandes : la commande 'node' et les 'path'. Toutes les autres commandes sont en fait l'une de ces deux commandes avec certaines options par défaut. Par exemple, la commande '\draw' correspond, en fait, à la commande '\path[draw]' (c'est à dire une commande 'path' avec l'option 'draw').

- **path** : Un *chemin* correspond à un enchaînement de segments droits, incurvés ou autres décrivant ainsi toute ou une partie de la figure.
- **node** : Un *nœud* correspond à un élément simple de la figure. C'est, par exemple, un rectangle, un cercle ou n'importe quelle forme simple avec un texte à l'intérieur ou pas. Les nœuds peuvent soit être placés à des endroits précis, soit faire parti d'un chemin (mais ne font alors pas partie du chemin, ils sont ajouté par dessus le chemin ensuite).

Voici un premier exemple simple de code en TikZ (à gauche le code et à droite le rendu) :

```
\begin{tikzpicture}
  \path[draw,thick] (1,0)--(0,0)--(0,1);
\end{tikzpicture}
```



Notez que nous aurions aussi pu écrire '\draw[thick]' à la place de '\path[draw,thick]'.

3.3.3 Votre premier schéma en TikZ

Le rendu final du premier schéma que nous allons réaliser est représenté sur la figure 5. Il s'agit juste d'une comparaison de courbes analytiques simples (mais en utilisant des couleurs). La première étape consiste à mettre en place la grille en utilisant un nœud '(left-down) grid (right-up)' :

```
\begin{tikzpicture}
  \draw[very thin,color=gray] (-.1,-1.1) grid (3.9,3.9);
\end{tikzpicture}
```

Notez que nous évitons de caler la grille sur des valeurs justes pour éviter d'afficher les bords. Ensuite, nous ajoutons les axes x et y à la figure en dessinant des chemins qui se terminent par une flèche :

```
\begin{tikzpicture}
  \draw[very thin,color=gray] (-.1,-1.1) grid (3.9,3.9);
  \draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
  \draw[->] (0,-1.2) -- (0,4.2) node[above] {$y$};
\end{tikzpicture}
```

Puis, finalement, nous ajoutons les courbes, avec les couleurs respectives (notez l'usage de la commande 'plot' et de l'ajout de la légende à travers l'ajout d'un nœud en fin de chemin) :


```

\begin{tikzpicture}
  \draw[very thin,color=gray] (-.1,-1.1) grid (3.9,3.9);
  \draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
  \draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
  \draw[color=gray] plot(\x,\x) node[right]{$f(x)=x$};
  \draw[color=blue] plot(\x,{sin(\x r)}) node[right]{$f(x)=\sin x$};
  \draw[color=red] plot(\x,{0.05*exp(\x)}) node[right]{$f(x)=\frac{1}{20} \mathrm{e}^x$};
\end{tikzpicture}

```

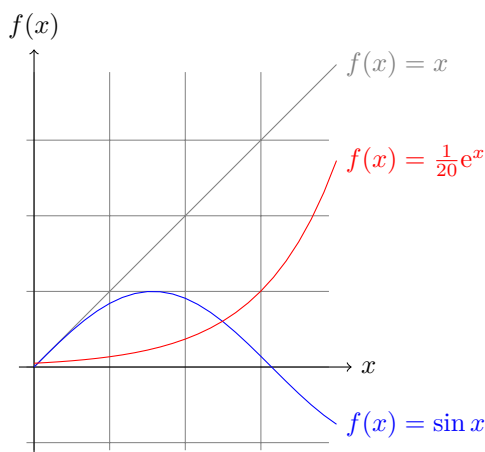


FIGURE 5 – Exemple de figure simple en TikZ.

Voilà, il ne reste plus qu'à intégrer le tout dans un environnement 'figure' et le tour est joué.

3.3.4 Quelques références et tutoriels

- *TikZ & PGF manual*, par Till TANTAU : <http://texdoc.net/pkg/tikz>
- *TikZ pour l'impatient*, par Gérard TISSEAU et Jacques DUMA : <http://math.et.info.free.fr/TikZ/bdd/TikZ-Impatient.pdf>
- *A very minimal introduction to Tikz*, par Jacques CRÉMER : <http://cremeronline.com/LaTeX/minimaltikz.pdf>
- *L^AT_EX and Friends : Creating Diagrams with TikZ*, par M. R. C. VAN DONGEN : <http://csweb.ucc.ie/~dongen/LAF/TikZ.pdf>
- Wikibook L^AT_EX : L^AT_EX/PGF/TikZ : <http://en.wikibooks.org/wiki/LaTeX/PGF/TikZ>
- Texamples.net, site rassemblant un bon nombre d'exemples en TikZ : <http://www.texample.net/>

Questions

1. Consultez les documents relatifs à TikZ, notamment [?, ?]. Vous y trouverez l'exemple représenté sur la figure 6. Puis, reproduisez à l'identique cette figure dans votre propre document.

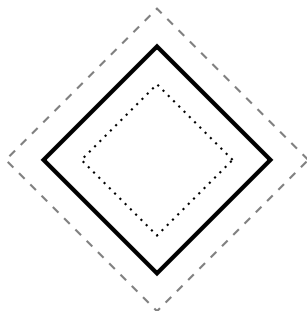


FIGURE 6 – Ceci est un exemple simple de figure TikZ.

2. L'exemple présenté sur la figure 7 se trouve quelque part dans le manuel de TikZ et PGF. Trouvez la figure et intégrez-la dans votre document.

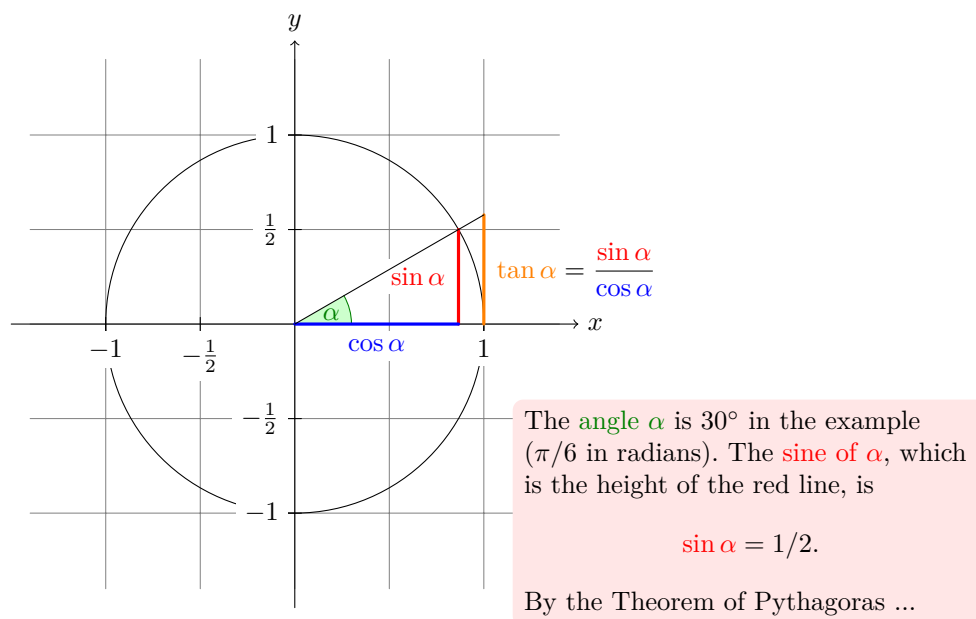


FIGURE 7 – Ceci est un autre exemple de figure TikZ.

3.4 Faire une présentation avec Beamer

Faire le rapport avec \LaTeX n'est qu'une partie du travail, très souvent vous aurez aussi à présenter vos résultats devant un public. Le module \LaTeX Beamer est justement là pour réaliser simplement des présentations vidéo-projetables. L'avantage d'utiliser \LaTeX à la fois pour le rapport et la présentation est que l'on peut aisément récupérer directement des parties du rapport et les utiliser dans la présentation (définitions, schémas, ...). C'est un gain de temps assez considérable, et le rendu final est aussi souvent bien meilleurs qu'avec les outils de présentations plus classiques. De plus, Beamer offre un langage assez simple à maîtriser et qui se combine parfaitement avec le module TikZ.

3.4.1 Votre première présentation Beamer

Utiliser le module Beamer nécessite l'utilisation d'une classe de document particulière :

```
\documentclass{beamer}
```

Une fois cette classe de document chargée, chaque diapo (ou *slide* en anglais) est représentée par un environnement 'frame'. Mis à part tout ceci, le document reste à peu près identique à ce que l'on retrouve dans un document \LaTeX . Voici un exemple complet qui vous servira de base pour le reste du tutoriel :

```
\documentclass{beamer}

%% Font packages
\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

%% Title, subtitle, authors, institute, date, ...
\title{My Slides Title}
\subtitle{My subtitle}
```

```

\author[Nom Prénom]{Nom Prénom\[-.25em]
\texttt{\scriptsize <email@address.org>}}

\institute[UBdx1, France]{Université Bordeaux~1, France}

\date{\today}

\begin{document}
  \begin{frame} % Title page
    \titlepage
  \end{frame}

  \begin{frame} % Table of contents
    \tableofcontents
  \end{frame}

  \section{This is the first slide}
  \begin{frame} % First slide
    \frametitle{This is the first slide}

    \vfill
    \begin{itemize}
      \item \textcolor{blue!75}{\bfseries This} is my first item;
      \item This is my second item;
      \item This is my last item.
    \end{itemize}
    \vfill
  \end{frame}

  \section{This is the second slide}
  \begin{frame} % Second slide
    \frametitle{This is the second slide}
    \framesubtitle{A bit more information about this}

    \begin{block}{My Block Title}
      This is the content of the block.
    \end{block}
    % More content goes here
  \end{frame}
\end{document}

```

L'exemple précédent comporte une page de titre, une table des matières ainsi que deux diapos. Tout ceci ressemble énormément au L^AT_EX classique, mais notez néanmoins quelques commandes spéciales. Par exemple, la première diapo contient une liste d'items encadrée par des '`\vfill`', il s'agit d'une commande qui ajuste de manière *élastique* l'espacement vertical en le remplissant avec des lignes blanches (comme si vous aviez collé deux élastiques de même force là où se situent les '`\vfill`'. De plus, vous pouvez aussi noter la commande '`\textcolor{blue!75}`' qui permet de colorer le texte qui suit avec une couleur bleue qui a une opacité de 75% (essayez de faire varier la couleur et le paramètre d'opacité et observez le résultat). Enfin, la seconde diapo utilise un `block` qui met en valeur du texte (définition, théorème, *etc*). D'autres types de `blocks` existent, essayez de remplacer '`block`' par '`alertblock`', puis par '`exampleblock`' et observez la différence.

Attention: Les `blocks` permettent de mettre en valeur certaines parties du texte, mais gardez les pour les cas où vous désirez faire ressortir une définition, un théorème, un exemple ou une remarque importante du reste du texte. À vouloir trop en mettre, vous finirez par obtenir l'effet contraire et

ne plus réussir à capter l'attention du public. Une bonne règle est de ne mettre qu'un seul **block** par diapo (à la rigueur vous pouvez mettre deux **blocks** à condition qu'ils soient de types différents).

3.4.2 Les thèmes Beamer

L'une des importante richesses de Beamer est de disposer d'un grand nombre de thèmes déjà prêts pour réaliser sa présentation. Le plus souvent, il s'agit de noms de villes (voir tableau 19 pour une liste complète) et on les charges avec la commande suivante, forcément placée avant le '`\begin{document}`' :

```
\usetheme{<theme name>}
```

Groupe 1	Groupe 2	Groupe 3	Groupe 4	Groupe 5
Bergen	Berlin	Berkeley	Copenhagen	Antibes
Boadilla	Ilmenau	Palo Alto	Luebeck	JuanLesPins
Madrid	Dresden	Goettingen	Malmoe	Montpellier
AnnArbor	Darmstadt	Hannover	Warsaw	
CambridgeUS	Frankfurt			
EastLansing	Singapore			
Pittsburgh	Szeged			
Rochester				

TABLE 19 – Les différents thèmes par défaut de Beamer (**Groupe 1**: sans barre de navigation; **Groupe 2**: avec une mini-frame de navigation; **Groupe 3**: avec table des matière sur le côté; **Groupe 4**: avec table des matières pour les sections/sous-sections; **Groupe 5**: avec barre de navigation arborescente).

3.4.3 Les colonnes

Une autre spécificité de la classe Beamer est l'environnement '`columns`' qui permet de faire une mise en page facile sur la diapo que vous affichez. Par exemple, voici une diapo en deux colonnes :

```
\begin{frame}
\frametitle{Test of the columns environment}

\begin{columns}[c]
\begin{column}{.45\linewidth}
This is the left side.
\end{column}
\begin{column}{.45\linewidth}
This is the right side.
\end{column}
\end{columns}
\end{frame}
```

Évidemment, rien ne vous empêche de mettre autant de colonnes que vous désirez. Vous pouvez même dépasser la largeur de la diapo (mais ce ne sera pas très esthétique). L'option '`c`' de l'environnement '`columns`' correspond à un alignement vertical centré des deux contenus. Il est possible aussi d'aligner les contenus des colonnes vers le haut '`t`'¹² (*top*), ou vers le bas '`b`' (*bottom*).

3.4.4 L'environnement verbatim

Il faut aussi savoir que les spécificités de la classe Beamer imposent aussi quelques contraintes. En effet, lorsque vous désirerez utiliser un environnement '`verbatim`' (ou du même genre, comme '`lstlisting`' et autres), il vous faudra protéger votre **frame** avec l'option '`fragile`' comme ceci :

12. L'option '`T`' est une alternative que vous pouvez essayer si '`t`' n'a pas marché comme vous le désirez.

```

\begin{frame}[fragile]
  \frametitle{Test of the verbatim environment}

  \begin{verbatim}
    Test !
  \end{verbatim}
\end{frame}

```

Si jamais vous oubliez de rajouter ‘fragile’, vous obtiendrez une erreur (souvent incompréhensible) au moment de la compilation.

3.4.5 Les overlays en Beamer

Pour finir avec les fonctionnalités particulières de Beamer, l’une des grandes puissances de cette classe repose définitivement sur les capacités à faire de l’*overlay*. C’est à dire à pouvoir faire apparaître et disparaître des parties de votre diapo afin de créer une animation permettant de suivre votre discours. Cette capacité marche avec du texte, mais prend toute sa puissance lorsqu’elle est couplée avec des figures TikZ, par exemple pour expliquer le fonctionnement d’un algorithme ou le schéma d’une preuve.

Commande	Usage
<code>\textbf<spec>{}</code>	Texte en gras si ‘spec’ est vraie
<code>\textit<spec>{}</code>	Texte en italique si ‘spec’ est vraie
<code>\emph<spec>{}</code>	Texte en emphase si ‘spec’ est vraie
<code>\alert<spec>{}</code>	Texte en rouge si ‘spec’ est vraie
<code>\item<spec></code>	Texte affiché si ‘spec’ est vraie
<code>\pause</code>	L’affichage s’arrête à cet endroit
<code>\only<spec>{}</code>	Texte apparaît si ‘spec’ est vraie
<code>\onslide<spec>{}</code>	Texte visible si ‘spec’ est vraie
<code>\uncover<spec>{}</code>	Texte visible si ‘spec’ est vraie
<code>\visible<spec>{}</code>	Texte visible si ‘spec’ est vraie
<code>\invisible<spec>{}</code>	Texte invisible si ‘spec’ est vraie
<code>\alt<spec>{alt1}{alt2}</code>	‘alt1’ si ‘spec’ est vraie, ‘alt2’ sinon

Spécification	Usage
<code><4></code>	Apparaît sur la page 4
<code><3-4></code>	Apparaît sur les pages 3 à 4
<code><-4></code>	Apparaît jusqu’à la page 4
<code><3-></code>	Apparaît à partir de la page 4

Modificateur	Usage
<code>\onslide</code>	visible/invisible
<code>\onslide+</code>	visible/invisible
<code>\onslide*</code>	apparaît/disparaît

TABLE 20 – Les différentes commandes d’*overlay* en Beamer.

Les spécifications des expressions sont très simples, il s’agit de définir les pages sur lesquelles le texte va apparaître (ou disparaître). De plus, il existe deux types de façons de faire disparaître du texte, soit il devient invisible (mais l’espace qu’il occupe reste vide), soit il disparaît complètement (et l’espace qu’il occupe est utilisé par le texte alentour). Pour mieux comprendre les différences entre visible/invisible et apparaît/disparaît, copiez et compilez le code suivant dans un **frame** :

```

\begin{itemize}
\item<-3> \onslide<2>{This} \uncover<2>{is} a test phrase.
\item \onslide+<2>{This} \invisible<2>{is} a test phrase.
\item \onslide*<2>{This} \only<2>{is} a test phrase.
\end{itemize}

```

3.4.6 Quelques références et tutoriels

— Wikibook *L^AT_EX : LaTeX/Presentations* <http://en.wikibooks.org/wiki/LaTeX/Presentations>

- *The beamer class : User Guide for version 3.27*, Til TANTAU, Joseph WRIGHT et Vedran MILETIĆ.
<http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf>
- *A Beamer Tutorial in Beamer*, par Charles T. BATTS, 2007.
<http://www.uncg.edu/cmp/reu/presentations/CharlesBatts-BeamerTutorial.pdf>
- *The Beamer class for L^AT_EX : A tutorial*, par Meik HELLMUND.
<http://www.mathematik.uni-leipzig.de/~hellmund/LaTeX/beamer2.pdf>
- *Making Slides ... and doing it with Beamer*, par RSI 2012 Staff, 2012.
<http://web.mit.edu/rsi/www/pdfs/beamer-tutorial.pdf>
- *Beamer Class Tutorial : Introduction to creating presentations in L^AT_EX using beamer class*, par Krzysztof FLESZAR, 2007. <http://staff.aub.edu.lb/~kf09/kfbeamertutorialen.pdf>

Questions

1. Faites quelques slides avec la classe `beamer` [?] qui contiennent :
 - Une page de titre ;
 - Une table des matières ;
 - Une liste itemize ;
 - Des `blocks` (normal, exemple et alerte) ;
 - Une diapo en double colonnes ;
 - Des overlays ;
 - Une image ;
 - Un environnement verbatim ;
 - Un schéma TikZ.
2. Essayez plusieurs thèmes sur votre exemple précédent.

4 Conseils d'écriture et de style

Écrire n'est pas chose aisée, car cet exercice demande beaucoup de pratique avant de pouvoir le faire correctement. Qui plus est, lorsqu'il s'agit d'un rapport ou d'une présentation scientifique, il est avant tout nécessaire de maîtriser et structurer les idées et concepts que vous désirez faire apparaître dans votre document. Tout ceci demande un recul et une expérience qui font souvent défaut au début.

Les sections qui suivent fournissent seulement quelques indices pour vous guider dans le processus d'écriture. Il ne s'agit en aucun cas de règles strictes à suivre aveuglement, mais des conseils à appliquer avec un nécessaire regard critique. N'hésitez pas aussi à consulter les différentes références qui sont données dans la dernière section.

4.1 Avant d'écrire !

La mise en forme du document final n'intervient que dans la dernière phase de votre travail. Même s'il est conseillé de rédiger des morceaux de votre document au fur et à mesure que vous avancez dans la compréhension du problème que vous traitez, la phase finale qui consiste à tout remettre en ordre et à se lancer dans la composition du document ne se fait lorsque vous aurez franchi un certain nombre d'étapes. Donc, avant de commencer à écrire, posez vous les questions suivantes :

- Avez-vous fait un état de l'art ?
- Connaissez-vous suffisamment votre sujet ?
- Quelle(s) idée(s) principale(s) voulez-vous faire passer ?
- À qui allez vous vous adresser à travers ce document ?
- Avez-vous un plan pour votre document ?
- Quelles sont les contraintes d'écriture qui vous sont imposées ?
(langue d'écriture, nombre de pages, informations à ne pas divulguer, ...)
- De combien de temps disposez-vous et quelle est votre efficacité journalière ?

4.2 Faites un plan détaillé et écrivez

La réalisation du plan est un point crucial dans l'élaboration du document. Mais, trop souvent, les plans proposés sont trop vagues. Un plan se doit vraiment de descendre jusqu'au niveau de la section, voire du paragraphe. Voici une ébauche d'algorithme qui vous permettra de constituer votre plan détaillé :

1. Faites un plan détaillé de tous les chapitres avec un résumé des idées que chaque chapitre contiendra.
2. En prenant le premier chapitre, faites un plan détaillé section par section avec un résumé des idées que vous voulez mettre dans chaque section.
3. Pour la première section, faites un plan détaillé paragraphe par paragraphe avec un résumé des idées que vous voulez mettre dans chaque paragraphe.
4. Écrivez la première section, puis passez à la section suivantes en appliquant le point précédent.
5. Une fois le premier chapitre fini, passez au suivant en revenant au point 2.
6. Phase finale : Assurez-vous qu'il y a une transition naturelle entre les différentes parties. Au besoin ajoutez quelques paragraphes, voire une section entière.

Attention: Cet algorithme convient à la plupart des gens, mais pas forcément tout le monde. Si vous avez l'habitude de faire autrement et que cela marche, ne changez pas de méthode.

4.3 Relisez, corrigez et retravaillez

Il arrive que des documents soient truffés de fautes d'orthographe, de typos, d'incohérences (phrases qui ne veulent rien dire, figures qui ne correspondent pas au texte, références à des chapitres qui n'existent pas, citations manquantes, ...). Ce genre de document ne donne pas envie de continuer la lecture. Ce ne sont peut-être que des détails mais ils s'avèrent extrêmement importants lorsque le lecteur doit mettre une note sur la qualité du travail et du document. Sans aller jusqu'à réclamer une orthographe parfaite, il est important de se relire (et de se faire relire par d'autres) et, au moins d'utiliser un outil de correction orthographique lorsque vous rédigez (comme `aspell`, `ispell`, ou encore `flyspell`). Il s'agit réellement d'un minimum que vous devez à vos lecteurs.

4.4 Que doit contenir un état de l'art ? Comment faire sa bibliographie ?

L'état de l'art est une partie importante d'un document, il montre à quel point vous vous êtes préoccupé de savoir ce qui a été fait avant vous, et montre aussi le recul que vous avez sur le sujet. Par exemple, arriver à relier votre sujet avec des sources qui n'ont, *a priori*, pas de rapport direct montre que vous avez une vision suffisamment abstraite de votre problème et des concepts sous-jacents pour vous permettre d'élargir votre horizon. Par contre, si vous citez seulement des documents qui ont un rapport évident avec votre sujet ou, pire, si vous citez des documents qui n'ont aucun rapport avec votre sujet, on en déduira que vous n'avez qu'une compréhension partielle de ce que vous faites.

Un état de l'art doit donc contenir l'ensemble des documents que vous pensez pertinents à lire et la raison pour laquelle il est pertinent de les lire, ceci afin que le lecteur puisse arriver à votre niveau d'expertise sur le sujet plus rapidement que vous ne l'avez fait vous-même. Vous devez donc écarter les documents qui vous ont fourvoyés, ou ceux qui pourraient être redondants avec d'autres, pour ne garder que les documents qui vous ont paru suffisamment synthétiques et qui vous ont permis de progresser dans le domaine que vous traitez.

La façon dont vous constituez votre bibliographie et dont vous citez vos références a aussi son importance. Le but d'une citation est avant tout de permettre au lecteur de retrouver facilement le document. Donc, plus vous mettez d'informations sur le document en question (titre, auteurs, date de publication, type de publication, éventuellement une URL qui ne risque pas de changer fréquemment, ...), plus le lecteur sera apte à retrouver le document en question et à en bénéficier.

Enfin, n'oubliez jamais qu'avec l'avènement d'Internet le nombre de documents et d'information qui transitent a littéralement explosé. Ce qui a pour effet de nous rendre extrêmement suspicieux lorsqu'un document a très peu de citations vers d'autres documents. Le plus souvent, cela veut juste dire que le ou les auteurs n'ont simplement pas fait le travail de recherche de l'existant comme ils auraient dû le faire.

4.5 Comment éviter le plagiat ?

Le plagiat est un mal qui touche tous les niveaux, de l'école primaire à la thèse (et au-delà). On le retrouve sous différentes formes, plus ou moins insidieuse, plus ou moins conscient. Et, la palette des types de plagiat est vaste et subtile avec une gravité des fautes qui varie de manière toute aussi subtile. Pourtant, les définitions classiques du terme de '*plagiat*', rassemblent tout dans un même sac :

« Le plagiat est une faute morale, civile, commerciale et/ou pénale consistant à copier un auteur ou créateur sans le dire, ou à fortement s'inspirer d'un modèle que l'on omet délibérément ou par négligence de désigner. »

Article Wikipedia sur le "Plagiat"¹³

Le copier/coller n'est qu'une forme basique de plagiat, car il s'agit aussi de citer "*l'origine de vos idées*", du moins, quand cela est possible. Sans aller jusqu'à l'extrémisme le plus avancé en matière de plagiat, le milieu scientifique repose justement sur les idées et notamment aussi sur le fait que la paternité d'une idée se doit d'être aussi clairement établie que possible. Il est donc normal que les enseignants-chercheurs, en tant que scientifiques, ne soient pas enclin à la clémence lorsqu'ils détectent des traces de plagiat dans des documents. Il faut donc à tout prix être aussi honnête que possible sur l'origine des textes, des images, des données et des idées que vous manipulez et d'en donner une source aussi précise que possible.

Attention: Une excuse classique pour justifier le plagiat est que la personne pense qu'elle n'arrivera pas à faire mieux. . . Mais, n'oubliez jamais que vous êtes justement là pour apprendre à faire mieux et ce n'est pas en recopiant ce que vous trouvez sur Internet que vous progresserez. Il faut pratiquer pour progresser.

4.6 Références utiles

- *Éléments de rédaction scientifique en informatique*, par Hadrien MÉLOT, 2008.
<http://informatique.umons.ac.be/algo/redacSci.pdf>
- *Quelques conseils pour ne pas rater sa présentation orale*, par Nicolas Seguin.
<https://www.ljll.math.upmc.fr/~seguin/conseils.php>
- *How to Write a Proof*, par Leslie Lamport, 1993.
<http://research.microsoft.com/users/lamport/pubs/lamport-how-to-write.pdf>
- *How to write proofs : a quick guide*, par Eugenia Cheng, 2004.
<http://cheng.staff.shef.ac.uk/proofguide/proofguide.pdf>
- *Le plagiat dans les systèmes éducatifs*, par François Daniel GIEZENDANNER, 2007.
<http://wwwedu.ge.ch/sem/documentation/documents/plagiat.pdf>
- *Qu'est-ce que le plagiat*, par Rémi BACHELET, 2012. <http://plagiat.ec-lille.fr/>
- *Voler des idées : Le plagiat*, par Rémi BACHELET, 2013.
http://plagiat.ec-lille.fr/Amphi_Plagiat.pdf

13. <http://fr.wikipedia.org/wiki/Plagiat>.