U.F.R. Sciences et Techniques
Année 2022 – 2023

UNIVERSITÉ DE TOULON

ECUE “LATX”

Édition scientifique avec LaTeX

Gloria Faccanoni

https://moodle.univ-tln.fr/course/view.php?id=7459

Dernière mise-à-jour : Samedi 21 janvier 2023

LaTeX est un système de préparation de documents qui occupe une position dominante parmi les scientifiques pour la réalisation de livres, d'articles de recherche, de présentations vidéo-projetées, de polycopiés de cours, de feuilles d'exercices, de notes de travail... Les raisons de son omniprésence sont, outre sa capacité à mettre convenablement en forme les formules mathématiques les plus compliquées, la qualité professionnelle du résultat (tant pour le texte que les mathématiques) permettant une publication directe, mais surtout sa façon de concevoir un document structuré, en séparant son fond (le sens du texte) de sa forme (la mise en pages) et en déchargeant l'auteur de tâches fastidieuses (comme les références croisées), lui permettant ainsi d'atteindre une grande productivité.

☞ Durée

Le module est organisé en 5 séances de cours-TP de 3h.

☞ Objectifs

- Se familiariser avec la rédaction scientifique et les bases de la composition avec LaTeX ,
- fournir un socle solide de connaissances de bases, si possible exempt de mauvaises habitudes,
- donner des pistes pour être en mesure par la suite de continuer seuls l'apprentissage.

☞ Programme pédagogique

1. *Prise de contact :* installation, structure d'un fichier source, types de document, structure du document, gestion automatique de la table des matières et des références.
2. *Mise en page :* support de la langue française, listes à pouce, énumérations, descriptions, tableaux et figures (flottantes), notes de bas de page...
3. *Mathématiques :* mise en forme de formules mathématiques, rédactions de théorèmes, exercices...
4. *Compléments :* gestion automatique de la bibliographie, présentation de codes, présentations vidéo-projetées, dessins avec LaTeX ...

☞ Validation

L'évaluation des connaissances consistera en la préparation à la maison d'un document libre de 6 à 10 pages (sans compter les pages blanches et les pages de titre ou de table des matières lorsque la classe `scrbook` ou `scrreprt` est utilisée). Les éléments obligatoires sont

- un titre (composé avec `\maketitle` ou l'environnement `{titlepage}`) et un abstract,
- une table des matières avec liens hypertextuels,
- des sections et sous-sections, des listes numérotées et non numérotées, des notes en bas de page et des notes à marge,
- des entêtes et pieds de page (qu'on personnalisera en utilisant soit le package `scrlayer-scrpage` dont on lira la documentation soit le package `fancyhdr`),
- des théorèmes et/ou exercices, exemples etc. (en utilisant un package, par exemple `amsthm`),
- des formules mathématiques *inline* et *display* sans numérotation, des formules avec numérotation et une référence à ces équations,
- des figures/schémas (réalisés par exemple avec le package `TikZ` ou tout autre package/logiciel de votre choix) dans un *float* de type `figure` avec *caption* et référence à cette figure,
- des tableaux dans des *float* de type `table` avec *caption* et référence à ce tableau,
- des commandes personnelles judicieusement choisies,
- une bibliographie (réalisée avec BibTeX ou BibLaTeX) et au moins une référence à une entrée de la bibliographie,
- un algorithme ou du code source (en utiliser un package spécifique).

Attention, seront considérés comme des fautes graves : utiliser de `$$...$$` au lieu de `\[...\]` ; utiliser de `\\` ou `\newline` en dehors d'un tableau; forcer la position d'un *float*; utiliser de `\newpage` ou `\clearpage` ou `\nocite*` etc.

Le but de ce document est de guider le nouvel utilisateur de LaTeX pour une prise en main efficace et, si possible, exempte de mauvaise habitudes. Attention, **il ne s'agit pas d'un manuel** mais d'un support aux cours/TP. Il est toujours en cours de rédaction, ne vous étonnez pas si vous découvrez des erreurs. Merci de me les communiquer.

Gloria FACCANONI

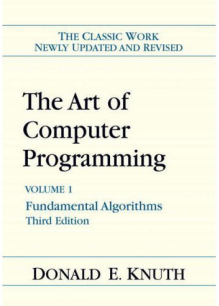
IMATH Bâtiment M-117
Université de Toulon
Avenue de l'université
83957 LA GARDE - FRANCE

☎ 0033 (0)4 83 16 66 72

✉ gloria.faccanoni@univ-tln.fr
🛈 http://faccanoni.univ-tln.fr

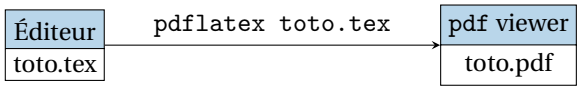
Table des matières

1 Introduction



En 1977, Donald Ervin KNUTH est en train d'éditer son livre *The Art of Computer Programming* mais il est mécontent du rendu obtenu. Il décide alors de créer son propre système de composition de texte, nommé TEX (prononcé *tek*, du grec ancien $\tau\varepsilon\chi\nu\eta$). Son utilisation étant assez ardue, un autre informaticien, Lesli LAMPORT, commence à développer un jeu de *packages* (ensemble de commandes) permettant d'accélérer la saisie de documents en TEX. Le mot LATEX (prononcé *latek*) est un jeu de mot basé sur TEX auquel sont rajouté les initiales du concepteur de LATEX .

Contrairement à d'autre logiciels, tels `OpenOffice` où l'on voit directement à l'écran ce que l'on tape, un document en LATEX est un fichier source qui doit être compilé avant d'être lisible par tous. Nous compilerons toujours avec `pdflatex` pour produire un fichier `pdf` (il y a d'autres formats de sortie et d'autres fichiers qui sont produits lors d'une compilation, mais on ne s'y intéressera pas dans ce document).



Contrairement aux logiciels de type WYSIWYG (*What You See Is What You Get*, ce que voyez est ce que vous obtenez), tel les documents produits avec `OpenOffice` ou `Microsoft Word`, LATEX sépare la forme du contenu et il demande au rédacteur de se concentrer uniquement sur la *structure logique* de son document et sur son contenu, tandis que la mise en page du document (césure des mots, alinéas, styles des titres, etc.) est laissée au logiciel. De ce fait, LATEX requiert un apprentissage initial plus important que celui qui est nécessaire pour les logiciels de type WYSIWYG, du moins pour la mise en page de petits documents simples. Mais une fois cette phase d'apprentissage accomplie, le fait de se concentrer sur le contenu et de laisser à LATEX le soin de présenter le document devient très appréciable : la qualité du document produit est élevée (respect des règles typographiques) et la gestion des références bibliographiques (avec une base de donnée BibTEX), les numérotations et la table des matières sont cohérentes sans qu'on ait à s'en soucier.

1.1 Pourquoi se mettre à LATEX

Apprendre à produire des documents avec LATEX demande un peu d'investissement, même s'il est possible d'apprendre au fur et à mesure de ses besoins. Cela vient du fait que c'est un système complètement différent des traitements de texte qu'on a l'habitude d'utiliser. Par exemple, pour écrire *différent* en italique, on n'a pas cliqué sur un bouton dans un menu comme on le ferait avec `OpenOffice`, mais en gros, on a saisi dans le fichier source quelque chose comme `\italique{différent}`. Tout ce document est construit selon ce principe, ensuite LATEX compile le fichier source et le transforme en un beau document `pdf`. À première vue, cela semble plus contraignant et moins convivial qu'un traitement de texte, mais alors pourquoi se mettre à LATEX?

— **Pour produire des documents scientifiques de qualité.** C'est pour cela que LATEX a été conçu au départ. On peut écrire facilement des formules mathématiques, dessiner des arbres, des molécules, des diagrammes commutatifs, de la musique, etc. Dans chaque communauté de scientifiques, certains ont créé des packages, disponibles sur Internet gratuitement et personnalisables, pour adapter LATEX à leurs besoins. Ici vous trouverez quelques exemples :

`http://www.tug.org/texshowcase/`

— **Pour la qualité typographique.** Les traitements de texte classiques n'ont pas été conçus avec l'expertise de typographes, contrairement au système LATEX . On en est assez vite convaincu à la vue d'un

document LaTeX : espace entre les caractères, césures, arrangement des paragraphes, mais également disposition des figures dans le texte, domaine pour lequel les traitements de texte sont très mal conçus.

— **Pour créer des gros documents.** C'est pour cette qualité de LaTeX que l'investissement est le plus rentable. On peut lui laisser la gestion de toutes les choses compliquées liées à la production de gros documents (livres, rapports de recherche, mémoires de stage, manuscrits de thèse…). En particulier,

  — il numérote automatiquement les sections, sous-sections, appendices, figures, formules, exercices, théorèmes, notes de bas de page, etc. ;

  — il crée tout seul la table des matières, la liste des figures et des tableaux et l'index ;

  — on peut numéroter très facilement les équations, les formules, les tableaux, les théorèmes, le livres dans la bibliographie… puis faire référence à ces numéros et à la page où ils apparaissent. Même si le document est retouché (par exemple, même si d'autres équations numérotées sont insérées à divers endroits dans le document), le document final restera cohérent ;

  — il gère très bien la disposition des figures et des tableaux dans un texte ;

  — on peut fusionner très facilement plusieurs documents, cela permet à plusieurs personnes de travailler sur des chapitres différents d'un même document. Par ailleurs, les documents LaTeX ne prennent que très peu de place sur le disque, contrairement aux documents produits par un traitement de texte.

— **Pour la pérennité.** C'est un critère déterminant pour un livre ou une thèse. Rien ne permet de dire qu'un document écrit avec `Microsoft Word` aujourd'hui puisse être parfaitement lisible (et modifiable) dans 10 ou 15 ans. Au gré des versions, des options disparaissent ou sont créées. Ce n'est pas le cas avec LaTeX . Les modifications qui y sont apportées ne se font jamais au détriment des utilisateurs.

— **Pour la souplesse.** Le principe même de LaTeX c'est un noyau commun, qui permet de créer tous les documents simples, et la possibilité de créer des nouveaux modules adaptés à des besoins particuliers. Concevoir ces modules demande beaucoup de talent en programmation mais dans chaque communauté ont été développées des bibliothèques spécifiques qui sont disponibles sur Internet : mathématiques, informatique, chimie, mais aussi partitions de musique, parties d'échecs, russe, grec, etc. On peut donc utiliser simplement LaTeX en se servant des modules créés par d'autres utilisateurs sans les concevoir soi-même. En revanche, il est très simple de créer des petites commandes adaptées à ses besoins. Par exemple, si j'en ai assez de saisir au clavier «Université de Toulon», je peux créer une commande `\UTLN`. À chaque fois que LaTeX va lire cette commande, il va automatiquement la traduire en «Université de Toulon».

— **Parce que c'est universel.** Pour échanger des documents produits avec LaTeX , on peut les transformer en fichier `pdf` ou même HTML. Peu importe que la personne avec qui on travaille utilise un PC avec Windows, Linux, Mac, Android. Et bien sûr, tout ça est entièrement gratuit.

1.2 Installation d'une distribution LaTeX , d'un éditeur dédié et d'un `pdf` viewer

Un des principaux avantages de LaTeX est qu'il ne nécessite pas grand chose pour pouvoir écrire des documents : une distribution LaTeX et un éditeur de texte. Certains éditeurs sont plus adaptés que d'autres et proposent des fonctionnalités plus ou moins intéressantes, telles que la coloration syntaxique, l'auto-complétion de certaines balises, etc. Durant cette formation, nous utiliserons l'éditeur de texte `Texmaker` qui est disponible sur les 3 plate-formes principales (Windows, Linux, Mac).

Installation sous Ubuntu

— Installation de la distribution `TeXlive` : dans *Synaptic* on fait une recherche en rentrant comme mot clé `texlive`.

— Installation de l'éditeur `Texmaker` : dans *Synaptic* on fait une recherche en rentrant comme mot clé `texmaker`.

— Installation d'un viewer `pdf` : dans *Synaptic* on fait une recherche en rentrant comme mot clé `Evince` ou `Okular` ou `Acroread`...

Installation sous Windows

— Télécharger le fichier ProTeXt à l'adresse `http://www.tug.org/protext/`

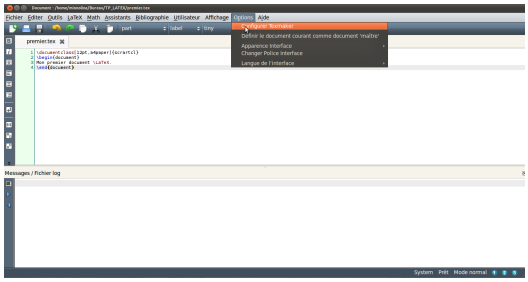
— Suivre les instructions pour l'installation de la distribution `MiKTeX` (choisir la version complète).

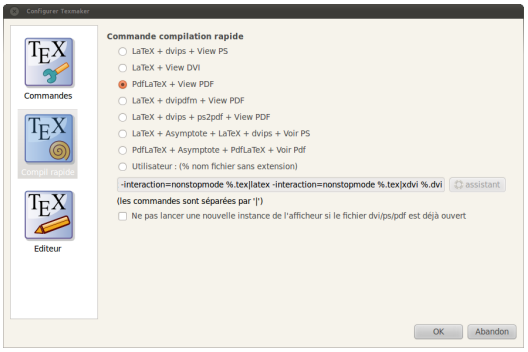
— Suivre les instructions pour l'installation de l'éditeur `Texmaker`.

— Installer un viewer `pdf` (par exemple `Acrobat Reader`).

1.3 Configuration de Texmaker

- Pour commencer on va démarrer notre éditeur de texte (sous Ubuntu : menu «Applications» → menu «Bureautique» → «Texmaker»). Une nouvelle fenêtre va s'ouvrir, c'est l'éditeur.
- Avant d'écrire le premier document, configurons un raccourci de l'éditeur pour qu'il fasse appelle à `pdflatex` et qu'il lance la visualisation du `pdf` automatiquement : menu «Option» → «Configurer Texmaker» ; une nouvelle fenêtre va s'ouvrir, dans cette fenêtre cliquer sur «Compilation rapide» et cocher «PdfLaTeX + View `pdf`».

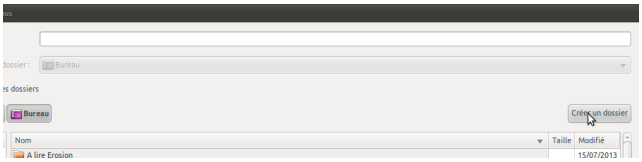




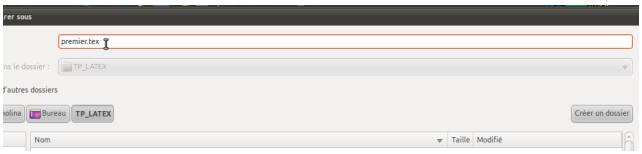
- Créons à présent un nouveau fichier (menu «Fichier» → «Nouveau») et copions le texte suivant :

```
\documentclass[12pt,a4paper]{scrartcl}
\begin{document}
Mon premier document \LaTeX.
\end{document}
```

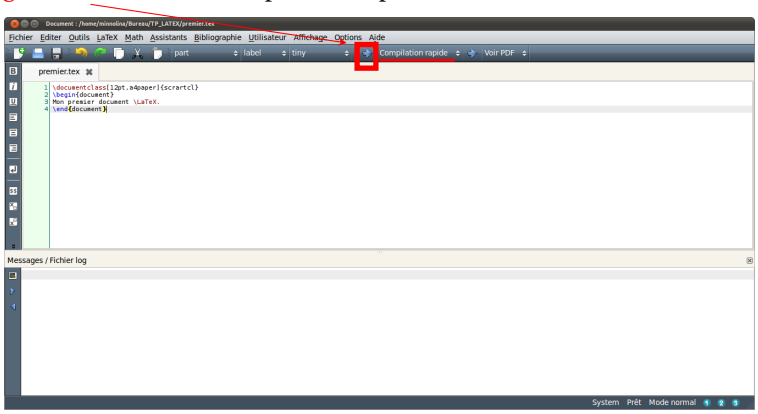
- Enregistrons ce document dans un répertoire `TP_LATEX` sous le nom `premier.tex` :
  — menu «Fichier» → «Enregistrer sous»,
  — on clique sur «Bureau» et on va créer un dossier en cliquant sur le bouton «Créer un dossier », on appelle ce dossier «`TP_LATEX`»



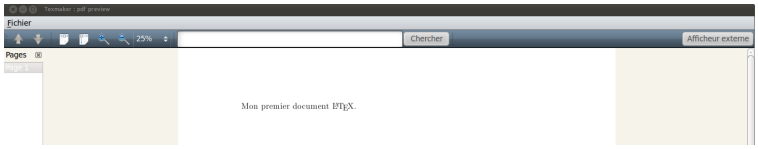
  — on appelle le fichier `premier.tex`



- Maintenant on va compiler notre fichier avec `pdflatex` et on lance la visualisation du `pdf` en cliquant sur la flèche à gauche du menu «Compilation rapide»



et on admire le résultat.



Un site de documentation pour `Texmaker` est disponible ici https://www.xm1math.net/texmaker/doc_fr.html

2 Premiers pas en LaTeX

Un fichier source LaTeX est un fichier `.tex` qui contient des balises. Ces éléments servent à structurer le document. Les balises sont introduites par la barre oblique inverse \ appelée *backslash*. La plupart des balises possède un argument (parfois plusieurs). Le cas échéant, l'argument est placé entre accolades `{ }`. Les balises peuvent aussi posséder des options placées entre crochets `[ ]` et séparées les unes des autres par des virgules.

Les instructions LaTeX se divisent en deux catégories :

— les commandes

```
\nomcommande[option_1,option_2,...]{arg_1}{arg_2}
```

comme par exemple `\documentclass[]{}`, `\tableofcontents`, `\maketitle`, `\section{}`, `\LaTeX` etc. Les commandes peuvent comporter plusieurs paires d'accolades, mais ne comportent pas toujours d'options.

— les environnements :

```
\begin{nom_envir}[option_1,option_2,...]

\end{nom_envir}
```

comme par exemple `\begin{document}...\end{document}`. Ici encore, les environnements ne comportent pas toujours d'options.

Les environnements peuvent être imbriqués, à condition que l'ordre de fermeture soit respecté :

```
\begin{aaa}
    \begin{bbb}
    ...
    \end{bbb}
    ...
\end{aaa}
```

Il existe une quantité inimaginable de commandes et environnements prédéfinis dans LaTeX (surtout après l'importation d'un package), voir par exemple https://ctan.org/pkg. Nous allons en voir une toute petite sélection parmi les plus utiles.

2.1 Structure de base d'un fichier LaTeX

Tout fichier LaTeX commence par la *commande*

```
\documentclass[opt_1,opt_2,...]{xxx}
```

On remarque que

- la commande commence par \ (*backslash*) qui s'obtient avec les touches `AltGr+8`,
- son nom est `documentclass`, à savoir la classe de document,
- ses arguments optionnels `opt_1,opt_2,...` sont entre
  - un crochet ouvrant `[` qui s'obtient avec les touches «`AltGr+5`»
  - et un crochet fermant `]` qui s'obtient avec les touches «`AltGr+)`».

Les commandes placées entre crochets sont des options. Dans tout ce cours on choisira pour `documentclass` les deux options `10pt,a4paper` pour utiliser une police de 10 points et un rendu sur une page de format A4.

- son argument `xxx` est entre
  - — une accolade ouvrante `{` qui s'obtient avec les touches «`AltGr+8`»
  - — et une accolade fermante `}` qui s'obtient avec les touches «`AltGr+=`».

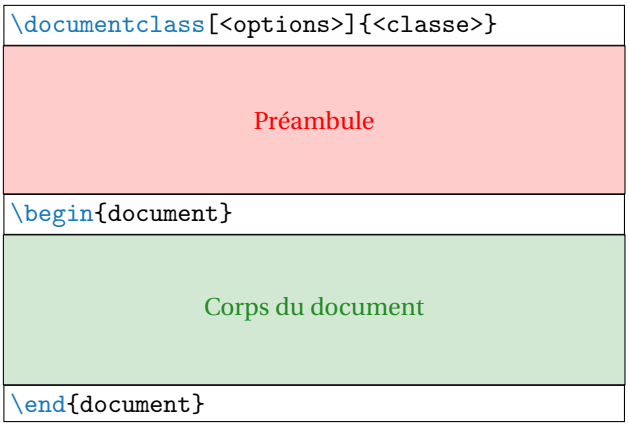
  Cette instruction dit à LaTeX que l'on souhaite utiliser une `class` de document (une mise en forme) particulière, nommée `xxx`. Dans tout ce cours on choisira `scrartcl` qui est la classe la plus courante pour des documents de quelques pages, mais d'autres choix sont possibles (`scrbook` pour des livres, `scrreprt` pour des rapports, `beamer` pour faire des présentations au vidéoprojecteur etc.). [1]

L'autre composante absolument obligatoire d'un document LaTeX est l'*environnement*

```
\begin{document}

\end{document}
```

Ce jeu de deux commandes sert à délimiter tout ce qui sera imprimé dans le 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. La partie entre `\begin{document}` et `\end{document}` est appelé le corps du document (c'est ici qu'on tape le texte qu'on veut voir apparaitre dans le `pdf`)



Voici un document LaTeX absolument minimal et le résultat après compilation :

```
\documentclass[10pt,a4paper]{scrartcl}
\begin{document}
Texte.
\end{document}
```

Texte.

[1] LaTeX a été écrit par un américain et est particulièrement adapté à cette langue. Pour pouvoir l'utiliser convenablement il nous faut charger des packages qui permettent de l'adapter à notre langue. C'est le cas du package `babel` et de son option `french` qui règle les problèmes linguistiques. Mais pour ce qui est de la mise en page, ce package ne règle pas le problème. En effet les classes standards (`article`, `book` et `report`) ont été conçues pour être imprimées sur le format de papier américain letter (279 × 216 mm) et legal (356 × 216 mm) et non notre format européen A4 (210 × 297 mm). C'est pour cette raison que les marges des documents LaTeX nous paraissent toujours trop grandes. Pour adapter la mise en page de LaTeX à nos standards européens, l'allemand Markus Kohm a développé KOMA-Script, qui est un ensemble de paquets mais surtout de classes qui remplacent les classes standard. C'est pour cela que dans ce document on utilise les classes `scrartcl`, `scrbook` et `scrreprt` au lieu des classes `article`, `book` et `report` décrites dans tous les manuels classiques.

2.2 Espaces dans le code source

Dans le fichier source, les espaces et les tabulations sont traitées indifféremment comme une «espace» par LaTeX . Plusieurs espaces consécutives sont considérées comme une seule espace. L'espace en début d'une ligne est ignorée. Une interruption de ligne est traitée comme une espace.

Une ligne vide entre deux lignes de texte marque la fin d'un paragraphe. Plusieurs lignes vides sont considérées comme une seule ligne vide.

```
Il n'est pas          important si on
met une ou plusieurs espaces.
Cette ligne appartient au même paragraphe,
pas la prochaine.

Une ligne vide démarre un nouveau
paragraphe.
```

Il n'est pas important si on met une ou plusieurs espaces. Cette ligne appartient au même paragraphe, pas la prochaine.
Une ligne vide démarre un nouveau paragraphe.

2.3 Commentaires

Dans un code, il est important d'annoter certaines parties. En LaTeX les commentaires se font à l'aide du symbole pour-cent `%` (lors de la compilation du document, les caractères situés après ce symbole seront ignorés).

2.4 Espaces entre alinéas dans le document final

On appelle *alinéa* la portion de texte comprise entre deux retours à la ligne, appelée aussi *paragraph*.

Le début d'une *alinéa* est indiquée par une indentation de la première ligne, c'est-à-dire une espace horizontale entre la marge de gauche et le premier mot du paragraphe. La raison est simple : si aucune indentation est utilisée, seule la longueur de la dernière ligne de l'alinéa précédent donnerait au lecteur un point de repère visuel. Mais dans certains cas il est très difficile de détecter si une ligne est plein ou non. Parfois on a alors envie d'augmenter l'espace vertical entre les alinéas. Mais ce type de repère visuel présente l'inconvénient de disparaître dans certains cas. Par exemple, après une formule centrée, il serait impossible de détecter si l'alinéa précédente se poursuit ou si une nouvelle alinéa commence. De même, quand on commence à lire en haut d'une nouvelle page, il peut être nécessaire de regarder la page précédente afin de déterminer si une nouvelle alinéa a été commencée ou non. Tous ces problèmes disparaissent lorsqu'on utilise l'indentation. *Une combinaison de l'indentation et de l'espacement vertical entre les alinéas est redondante et doit donc être évitée.* L'indentation est parfaitement suffisante par elle-même. Le seul inconvénient de l'indentation est la réduction de la longueur de ligne. L'utilisation de l'espacement entre les alinéas est donc justifiée lors de l'utilisation de lignes courtes, par exemple dans un journal.

Indépendamment de l'explication ci-dessus, il y a souvent des demandes pour une mise en page de documents avec **un espacement vertical entre les alinéas au lieu de l'indentation**. La classe KOMA-Script fournit un grand nombre d'options pour gérer cet espace vertical : `parskip=full`, `parskip=full-`, `parskip=full*`, `parskip=full+` et `parskip=half`, `parskip=half-`, `parskip=half*`, `parskip=half+`. Les quatre options `full` définissent chacune un espacement vertical entre les alinéas d'une ligne. Les quatre options `half` définissent un espacement vertical d'une demi-ligne. Afin d'éviter qu'un changement d'alinéa passe inaperçu, par exemple après un saut de page, trois options (-, * et +) veillent à ce que la dernière ligne d'une alinéa ne soit pas complètement remplie. Si on met -, LaTeX ne fera rien (la dernière ligne de l'alinéa précédente pourra être remplie) ; si on ne met rien, il laissera un espace de longueur `1em` en fin de ligne ; si on met + il laissera un tiers de la ligne libre et enfin, si on met *, il laissera au moins un quart de la ligne libre.

☞ Avertissement sur la personnalisation d'un document

Le but de la typographie est de rendre un document beau et agréable à lire. Pour cela, il ne faut pas que des éléments typographique détournent le lecteur du fond, ni que la mise en page n'entraîne de *fatigue visuelle*. Par exemple une règle couramment admise est que la longueur d'une ligne de texte (en typographie on dit la justification) ne doit pas dépasser 60 à 70 caractères (espace compris). Au-delà l'œil fatigue et il est plus difficile de localiser la prochaine ligne. Pour un texte écrit sur une seule colonne, une justification de 65 caractères est considérée comme idéale. C'est pour éviter la fatigue visuelle que les journaux, qui cherchent pour des raisons d'économie à mettre le plus de texte par page, sont écrit en colonnes pour conserver une justification acceptable. Donc si on décide de modifier la taille du texte, il faut essayer de respecter le plus possible cette règle.

Le *gris typographique* est l'apparence grise que prend le corps de texte, quand on le regarde de loin ou si on plisse les yeux pour voir le texte flou. Plus ce gris est homogène, plus la lecture du texte sera facile et agréable. Au contraire, quand l'homogénéité de ce gris est rompue par un mauvais interlignage, par des «rivières» ou par des «lézardes» (les lignes blanches composées d'espaces, les unes sous les autres, qui traversent les textes mal mis en page), l'attention portée au texte diminue car ces défauts détournent du texte et rendent la lecture difficile. Obtenir un bon gris n'est pas facile, il dépend de la police utilisée, de sa taille, de la justification, de l'interligne…

Avant de se lancer dans des modifications du comportement standard de LaTeX , il faut bien comprendre que la typographie est une science difficile. Il aurait été plus facile d'écrire un logiciel laissant à l'utilisateur le choix des réglages typographiques (comme la dimension des marges par exemple). Mais les résultats sont souvent calamiteux, car les typographes improvisés que nous sommes ne savent pas ce qui est important ou incorrect, d'autant plus que nous nous sommes habitués à une typographie pauvre. Avec l'avènement des ordinateurs et des traitements de texte, on voit fleurir dans les textes tout un tas de fioritures (changement de fontes, de corps, de casse,…) censées faire beau, mais c'est oublier que la typographie est avant tout au service du texte, que le lecteur doit rester concentré, sans fatiguer.

Si j'ai tenu à évoquer ces questions d'ordre typographique, c'est pour comprendre que l'on ne s'improvise pas typographe et que **dans la plupart des cas vouloir modifier les réglages proposés par défaut par LaTeX est une mauvaise idée**. Pour fabriquer le texte, LaTeX utilise des algorithmes qui respectent les règles typographiques. Même si, par la suite, on verra comment régler certaines des paramètres de la mise en page, je conseille de le faire avec parcimonie et plutôt de faire confiance à LaTeX .

2.5 Le système de packages de LaTeX

LaTeX dispose d'un mécanisme d'importation de packages qui permet d'étendre ses possibilités. La syntaxe pour importer un package est

```
\usepackage[option_1,option_2,...]{nom_du_package}
```

qu'il faut mettre dans le préambule (les options des packages utilisent des caractères non accentués). Voici le fichier source avec le préambule de base qu'on utilisera pour nos documents :

```
\documentclass[10pt,a4paper]{scrartcl}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{textcomp}
\usepackage{amsmath,amssymb}
\usepackage{lmodern}
\usepackage{graphicx}
\usepackage[dvipsnames,svgnames]{xcolor}
\usepackage{microtype}
\usepackage{hyperref} \hypersetup{colorlinks=true,linkcolor=Brown,urlcolor=Navy,
breaklinks=true,bookmarks=true,pdfstartview=XYZ}
```

```
\begin{document}
Test!
\end{document}
```

Regardons plus en détail chacun de ces packages.

- `\usepackage[utf8]{inputenc}`
  Ce package déclare l'encodage du fichier source. Il est chargé avec l'option `utf8` pour dire à LaTeX que le document source sera encodé en UTF8. On fait ce choix car c'est le codage par défaut des caractères sous Ubuntu, mais si vous souhaitez créer un document sur un système Windows par exemple, il faudra remplacer cette option par `latin1`.
- `\usepackage[T1]{fontenc}` Ce package, qui est chargé avec l'option T1, s'occupe de la gestion des accents (pour les pdf). 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[french]{babel}`
  La langue naturel de LaTeX est l'anglais. Le rôle du package `babel` est de permettre à LaTeX de parler d'autres langues, à l'aide d'options. Chaque langue possède des règles typographiques qui lui sont propre. On ne coupe pas les mots de la même manière en anglais et en français. Les items d'une listes commencent par • en anglais et — en français. En anglais et en allemand on ne met pas d'espace avant les signes doubles (`;` `:` `?` `!`) en français on met une espace insécable avant et une espace normale après pour le `:` et pour les `;` `?` `!` on met avant une espace fine insécable et une espace normale après.
  Pour charger la prise en compte de la langue française, on utilise l'option `french`. 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); 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,french]{babel}`. L'option `french` permet de respecter toutes les subtilités de la typographie française, pour la traduction automatique des mots (par exemple le mot «chapter» sera traduit automatiquement en «chapitre»), les caractères spécifiques à une langue (comme les guillemets français), etc.
- `\usepackage{textcomp}`
  Il permet l'accès à certains caractères supplémentaires, non disponibles dans l'encodage `T1` utilisé par `fontenc`.
- `\usepackage{amsmath,amssymb}`
  Ils sont utiles pour rédiger des formules mathématiques (on reviendra dessus au chapitre dédié aux mathématiques). `amssymb` définit nombreux symboles, `amsmath` est pratiquement indispensable. Recommandation : ~~`amsmath`~~ ⇝ `mathtools` : ce package charge `amsmath` en sous-main et en corrige quelques bogues et limitations.
- `\usepackage{lmodern}`
  Ici on fait 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` ou `pxfonts` (Palatino) ou encore `txfonts` (Times), `kpfonts` etc.
- `\usepackage{graphicx}`
  Il permet d'inclure des images dans le pdf. Recommandation : ~~`graphicx`~~ ⇝ `graphbox` : ce package charge `graphicx` en sous-main et ajoute des options pour l'alignement vertical (utile pour des images sur la même ligne).
- `\usepackage[dvipsnames,svgnames]{xcolor}`
  Il permet d'utiliser les couleurs (voir la documentation https://www.ctan.org/pkg/xcolor).

- `\usepackage{microtype}`
  Il introduit des améliorations typographiques (toujours charger `microtype` **après** un package changeant de fonte comme `lmodern` ou `fourier`).
- `\usepackage{hyperref}`
  Le package `hyperref` (toujours à mettre **en dernier** même si on rajoute des packages supplémentaires), 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; avec l'option `colorlinks=true, linkcolor=Brown` on n'a plus de cadre mais les entrés sont écrite en `Brown`). On termine par dire à `hyperref` qu'on autorise les coupures des URL et d'utiliser le zoom standard (par défaut, `hyperref` change le zoom à quelque chose de trop petit pour être lu) : `\hypersetup{pdfstartview=XYZ}`

2.5.1 Pour aller plus loin

Il existe un grand nombre de packages LaTeX sur CTAN pour différentes disciplines et exigences. La liste complètes des packages disponibles ainsi que leur manuel d'utilisation est disponible à l'adresse

https://ctan.org/

Cependant, la plupart de ces packages ne doivent pas être installés directement et manuellement à partir de CTAN. Au lieu de cela, vous devez utiliser le gestionnaire de packages de la distribution LaTeX que vous utilisez pour l'installation.

Avec `TeXlive` (la distribution classique sous Ubuntu), pour accéder rapidement à la documentation d'un package on peut taper dans un terminal (qu'on peut ouvrir avec la combinaison de touches «ctrl+alt+T»)

```
texdoc nomDuPackage
```

Par exemple, pour obtenir la documentation de `xcolor` on tapera

```
texdoc biblatex
```

2.6 Caractères spéciaux

Comme nous l'avons vu précédemment, les commentaires sont précédés du symbole %. Ainsi, pour écrire se symbole, il faudra utiliser une syntaxe indirecte, comme pour d'autres symboles dont voici une liste (non exhaustive) :

- le *backslash* \ pour commencer une commande. Pour afficher le *backslash* on utilise la commande `\textbackslash`.
- les accolades { et } pour définir des groupes. Pour afficher les accolades on utilise les commandes \{ et \}.
- le pour-cent % pour commencer un commentaire (lors de la compilation du document, les caractères situés après ce symbole seront ignorés). Pour afficher le pour-cent on utilise la commande \%.
- un ou plusieurs dollars $ pour passer en mode mathématique. Pour afficher un dollar on utilise la commande \$.
- l'*underscore* _ pour écrire des indices en mode mathématique. Pour afficher l'*underscore* on utilise la commande _.
- le chapeau ^ pour écrire des exposants en mode mathématique. Pour afficher le chapeau on utilise la commande `\textasciicircum`.
- l'*ampersand* & pour séparer les colonnes d'un tableau. Pour afficher l'*ampersand* on utilise la commande \&.

- le *hash* # pour définir des nouveau commandes. Pour afficher le hash on utilise la commande \#.
- le tilde ~ pour produire une espace insécable.[2] Pour afficher le tilde on utilise la commande \textasciitilde.

2.7 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 à la figure 2.1. Analysons la *structure* de ce document. Il 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 une section non numéroté (et qui n'est pas dans la table des matières) dont le titre est «Introduction», du texte, puis une section numérotée dont le titre est «Rappels», du texte, une sous-section numérotée dont le titre est «Condition initiale», du texte, une sous-section numérotée dont le titre est «Problème de Cauchy» puis du texte et pour finir une section numérotée dont le titre est «Exercices».

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 «Équations différentielles» 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.

Voici les commandes nécessaires pour générer ces éléments.

Dans le préambule :

- Les 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{Équations différentielles}
\author{Gloria Faccanoni}
\date{15 janvier 2014}
```

  Si on ne met pas la ligne avec \date (par exemple on met un % devant l'instruction), la date choisie sera la date du jour de compilation du fichier.

  Si on commente la ligne qui importe le paquet babel, la date sera en anglais.

- Pour afficher le texte de l'exemple sans devoir le taper on peut charger dans le préambule le package lipsum et puis utiliser la commande \lipsum. Cette commande génère par défaut les paragraphes 1 à 7 du Lipsum.[3] Pour générer un Lipsum contenant les 2 premiers paragraphes, on écrira \lipsum[1-2]. Pour afficher la première phrase du deuxième paragraphe, on écrira \lipsum[2][1]. Le paquet blindtext propose un résultat similaire.

Dans le corps du document :

- Après le \begin{document} (c'est-à-dire dans le corps du document), on utilise \maketitle pour imprimer le titre, l'auteur et la date (cette commande ne prend pas d'argument, contrairement aux trois précédentes). Rappelons que ces informations ont été mentionnées dans le préambule. Si la commande \maketitle est omise, le document obtenu après la compilation n'inclura pas ces informations.

[2] L'espace insécable est une espace qui ne peut être coupée par un saut de ligne. Elle s'écrit ~ en LaTeX (\, pour l'espace fine insécable).

[3] Le *Lorem Ipsum* est simplement du faux texte employé dans la composition et la mise en page avant impression. C'est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du *Lorem Ipsum* sur un texte générique comme «bla bla bla» est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du *Lorem Ipsum* leur faux texte par défaut, et une recherche pour «Lorem Ipsum» vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction.

Équations différentielles

Gloria Faccanoni

15 janvier 2014

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Table des matières

Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer 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 Rappels

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.

1.1 Condition initiale

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.

1.2 Problème de Cauchy

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

2 Exercices

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

FIGURE 2.1 – Exemple de document de deux pages avec titre, table des matières et sections

- Lors d'une publication scientifique (notamment des articles), il est usuel de démarrer celle-ci avec un résumé (*abstract* en anglais), censé donner au lecteur une vue d'ensemble de ce qu'il doit attendre du document. La classe `scrartcl` fournit un environnement `abstract` à cette fin :

```
\begin{abstract}
...
\end{abstract}
```

- Pour afficher la table des matières, on utilise la commande

```
\tableofcontents
```

  Comme `\maketitle`, cette commande ne prend pas d'argument. LaTeX permet de générer automatiquement la table des matières. Pour que la table soit générée, **il faut compiler le document deux fois**. En effet, la première compilation créera un fichier (dont l'extension est `.toc`) dans lequel seront mises les diverses informations à insérer dans le sommaire) et la seconde compilation lira le contenu de ce fichier.

- 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}
```

  Les sections non numérotées n'apparaissent pas dans la table des matières. Pour les ajouter, au lieu de la commande `\section*` on utilisera la commande

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

  Pour faire une sous-section, on utilise `\subsection` (la syntaxe est la même que pour `\section`). Dans la classe `scrartcl`, les commandes de sectionnement suivantes sont disponibles :

  `\part` et `\part*` et `\addpart`
  `\section` et `\section*` et `\addsec`
  `\subsection` et `\subsection*`
  `\subsubsection` et `\subsubsection*`
  `\minisec`
  `\paragraph`
  `\subparagraph`

  Dans les classes `scrreprt` et `scrbook`, une commande de sectionnement entre `\part` et `\section` est disponible :

  `\chapter` et `\chapter*` et `\addchap`

  Attention : les commandes `\addpart`, `\addchap`, `\addsec` et `\minisec` ne sont disponibles qu'avec les classes `scrartcl`. `scrbook`, `scrreport`)

Exercice 1
Reproduire le document de la figure 2.1.

2.8 Inclusion de fichiers LaTeX

Quand un fichier `.tex` commence à devenir long, il est souvent utile de pouvoir en compiler une seule partie (par exemple juste un chapitre ou une section). Il est donc intéressant de séparer le fichier source en plusieurs fichiers.

Deux commandes permettent d'effectuer ce travail : les commandes `input` et `include`. Elles prennent un unique paramètre qui est le nom du fichier à inclure. Attention : avec `input` il faut ajouter l'extension

dans le nom du fichier à importer, avec `include` il ne faut pas indiquer l'extension. De plus, la commande `include` démarre une nouvelle page avant d'inclure le contenu du fichier.

Fichier `ch1.tex`

```
\section{Mon chapitre}
Bla bla
```

Fichier `ch2.tex`

```
\section{Un chapitre}
Bli bli
```

Fichier principale à compiler

```
\documentclass{scrartcl}

\begin{document}

\input{ch1.tex} % ou \include{ch1} NB sans extension
%\input{ch2.tex} non importé car ligne commentée

\end{document}
```

Le fichier `ch1.tex` n'est absolument pas un fichier LaTeX complet. Lorsque LaTeX rencontre la commande `input`, il va ouvrir le fichier `ch1.tex` et simplement copier-coller son contenu à la place de `\input{ch1}` comme si on avait écrit le document en un seul fichier comme ci-dessous :

Fichier unique

```
\documentclass{scrartcl}

\begin{document}

\section{Mon premier chapitre}
Bla bla

\end{document}
```

Pour aller plus loin on pourra utiliser (après avoir lu la documentation) le package `subfiles` disponible à l'adresse https://ctan.org/pkg/subfiles

2.9 Étiquettes, références croisées et liens externes

Il est parfois utile dans un document de faire référence à des parties du texte (une section, une équation, un théorème, une entrée dans la bibliographie etc.). Plutôt que d'indiquer à la main les pages concernées (opérations hasardeuse par ailleurs, car le numéro de page peut changer en cours de rédaction), il est plus simple de laisser LaTeX gérer ces opérations. Ainsi, on dispose des trois commandes suivantes :

- la commande `\label{eti}` insère une étiquette invisible, nommée `eti` ;
- `\ref{eti}` affiche la référence à l'étiquette `eti` ;
- `\pageref{eti}` insère le numéro de la page où se situe l'étiquette `eti`.

Comme tout ce qui est dynamique, il faut deux compilations pour que cela s'af?che correctement. Si on a chargé dans le préambule le package `hyperref`, la référence est cliquable et renvoi à la page où on a mis l'étiquette invisible. Pour que la référence soit correcte il faut compiler **deux** fois.

Voici un exemple

1 Introduction

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

1.1 Premier Essai

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

1.2 Deuxième Essai

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

On a vu à la section 1.1 à la page 1 que bla bla bla

obtenu avec le code suivant :

```
\documentclass[a4paper,10pt]{scrartcl}

\usepackage[latin1]{inputenc} % sous Windows
% \usepackage[utf8]{inputenc} % sous Linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage[dvipsnames,svgnames]{xcolor}
\usepackage{hyperref}\hypersetup{colorlinks=true,linkcolor=Brown,citecolor=
ForestGreen}

\begin{document}
\section{Introduction}
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla

\subsection{Premier Essai}\label{subsec.toto}
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla

\subsection{Deuxième Essai}
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 bla bla bla

On a vu à la section~\ref{subsec.toto} à la page~\pageref{subsec.toto} que bla bla
bla
\end{document}
```

Pour imprimer les étiquettes dans le pdf (pendant la rédaction), il suffit d'importer dans le préambule le paquet showkeys : `\usepackage{showkeys}`

Pour afficher un lien internet (cliquable pour aller à la page web voulue), on utilise la commande `\url{}` ou la commande `\href{}{}` du package `hyperref` :

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

\href{http://www.google.com}{Google}
```

http://www.google.com
Google

2.10 Notes de bas de page ou dans la marge

La commande `\footnote` imprime une note de bas de page en bas de la page en cours. Les notes de bas de page doivent être placées après le mot où la phrase auquel elles se réfèrent. [4] Les notes qui se réfèrent à une (partie de) phrase devraient être placées après une virgule ou un point. [5]

L'alinéa précédente a été obtenue par le code

```
... elles se réfèrent.\footnote{La typographie française demande ... du document.}
Les notes qui se réfèrent à une (partie de) phrase devraient être placées après une
 virgule ou un point.\footnote{Remarquez que les notes de bas de page détournent l'
attention du lecteur du corps du document.} La note ...
```

Exemple de note dans la marge

La commande `\marginpar` imprime une note dans la marge de la page en cours. Cette phrase a été obtenue par le code

```
La commande \marginpar imprime une note dans la marge\marginpar{Exemple de note
dans la marge} de la page en cours.
```

2.11 Listes à puce, énumérations et descriptions

Une manière de structurer le texte à l'intérieur d'un paragraphe est d'utiliser les listes.

- L'environnement `itemize` est utilisé pour des listes à puces :

```
Liste des courses:
\begin{itemize}
\item pain
\item lait
\item farine
\end{itemize}
```

Liste des courses :
- — pain
- — lait
- — farine

Pour redéfinir une seule label d'une listes à puce, on pourra écrire :

```
Liste des courses:
\begin{itemize}
\item pain
\item[$\bullet$] lait
\item farine
\end{itemize}
```

Liste des courses :
- — pain
- • lait
- — farine

Pour redéfinir toutes les labels des listes à puce pour avoir des •, on pourra écrire dans le préambule juste après le `\usepackage[french]{babel}` :

```
\frenchsetup{StandardItemLabels=true}
```

- L'environnement `enumerate` est utilisé pour des énumérations :

```
Risotto:
\begin{enumerate}
\item faire ruisseler l'oignon et le
beurre,
\item ajouter du vin blanc et le riz et
laisser évaporer,
\item ajouter le bouillon peu à peu,
\item \dots
\end{enumerate}
```

Risotto :
1. faire ruisseler l'oignon et le beurre,
2. ajouter du vin blanc et le riz et laisser évaporer,
3. ajouter le bouillon peu à peu,
4. ...

[4] La typographie française demande une espace fine avant la marque de renvoi à la note. Celle-ci est insérée automatiquement par le package `babel` si le français est la langue principale du document.

[5] Remarquez que les notes de bas de page détournent l'attention du lecteur du corps du document.

- L'environnement `description` est utilisé pour des descriptions :

```
ECUE Compétences:
\begin{description}
\item[Titre:] Édition scientifique
              avec \LaTeX
\item[Durée:] 12h
\end{description}
```

ECUE Compétences :

Titre : Édition scientifique avec LaTeX

Durée : 12h

On peut imbriquer différents types de listes :

```
\begin{itemize}
\item À faire:
    \begin{enumerate}
    \item Sortir le chien
    \item Rentrer les poubelles
    \item Préparer à manger
    \end{enumerate}
\item Ce qui sera fait:
    \begin{enumerate}
    \item Prendre une bière
    \item Allumer la TV
    \item Dormir
    \end{enumerate}
\end{itemize}
```

- À faire :
  1. Sortir le chien
  2. Rentrer les poubelles
  3. Préparer à manger
- Ce qui sera fait :
  1. Prendre une bière
  2. Allumer la TV
  3. Dormir

Exercice 2

Reproduire le document suivant :

1. Ont des ailes
   a) Volent
      i. Hirondelle
      ii. Aigle
      iii. Ara
   b) Ne volent pas
      i. Poule
      ii. Autruche
2. Vivent dans la mer
   - Respirent dans l'eau
     - requin
     - hippocampe
   - Respirent hors de l'eau
     - baleine
     - dauphin
3. Vivent sur terre

   Australie : kangourou, koala

   Europe : loup

Pour personnaliser les listes on pourra utiliser (après avoir lu la documentation) :

— soit le package `enumitem` disponible à l'adresse https://ctan.org/pkg/enumitem

— soit le package `paralist` disponible à l'adresse https://ctan.org/pkg/paralist

Exercice 3

Reprendre l'exercice 2 en ajoutant dans le préambule `\usepackage[pointedenum]{paralist}`. Qu'est-ce qui change ?

2.12 Afficher des codes sources avec `listings`

Pour insérer du code source dans un document LaTeX , il est possible d'utiliser le package `listings`. On va également pouvoir préciser le langage qui doit être utilisé pour la coloration ou encore la coloration du fond.

Le package listings propose la commande `\lstinline` pour des codes sources dans le texte, l'environnement `{lstlisting}` pour imprimer plusieurs lignes de code et une commande `\lstinputlisting` qui permet d'inclure un fichier externe.

Dans le préambule on écrira par exemple

```
\usepackage{listings}
\lstset{
    upquote = true,
    columns = flexible,
    basicstyle = \ttfamily,
    language = python,
    backgroundcolor = \color{Navy!10},
    keywordstyle = \color{ForestGreen}\bfseries,
    % numbers=left, numberstyle=\tiny,
    tabsize = 4,
}
```

Puis dans le corps du document on écrira soit

```
la fonction \lstinline!print()! ...
```

soit

```
\begin{lstlisting}
impaires = lambda L,R : [i for i in range(L,R+1) if i%2!=0]
print(impaires(1,15))
\end{lstlisting}
```

soit

```
\lstinputlisting{source.py}
```

Cette dernière instruction inclut directement le fichier source `.py`. Voici un exemple de résultat après importation d'un fichier python :

```
impaires = lambda L,R : [i for i in range(L,R+1) if i%2!=0]
print(impaires(1,15))
```

Comme d'habitude, pour plus d'informations il faut lire la documentation officielle :

https://ctan.org/pkg/listings

Voir aussi

https://borntocode.fr/latex-comment-inserer-et-customiser-du-code-source/

Problème : lorsqu'on utilise l'encodage UTF8 il ne faut pas mettre de lettres accentuées dans ces codes sources. On a lors trois choix : soit on renonce à l'UTF8 dans tout le document, soit on renonce au caractères accentués dans nos codes, soit une stratégie intermédiaire qui va corriger ce problème pour les importations via `lstinputlisting`. Pour cela il faut remplacer le package `listings` par le package `listingsutf8` qui n'est autre qu'un patch à listings :

https://ctan.org/pkg/listingsutf8

.

```
\usepackage{listingsutf8}
\lstset{
    upquote = true,
    columns = flexible,
    basicstyle = \ttfamily,
    language = python,
    backgroundcolor = \color{Navy!10},
    keywordstyle = \color{ForestGreen}\bfseries,
    % numbers=left, numberstyle=\tiny,
    tabsize = 4,
    % NEW
    inputencoding = utf8, % latin1
    extendedchars=false,
    literate = {é}{{\'e}}{1}%
             {è}{{\`e}}{1}%
             {à}{{\`a}}{1}%
             {â}{{\^a}}{1}%
             {ç}{{\c{c}}}{1}%
             {œ}{{\oe}}{1}%
             {ù}{{\`u}}{1}%
             {É}{{\'E}}{1}%
             {È}{{\`E}}{1}%
             {À}{{\`A}}{1}%
             {Ç}{{\c{C}}}{1}%
             {Œ}{{\OE}}{1}%
             {Ê}{{\^E}}{1}%
             {ê}{{\^e}}{1}%
             {î}{{\^i}}{1}%
             {ï}{{\"i}}{1}%
             {ô}{{\^o}}{1}%
             {û}{{\^u}}{1}%
}
```

`inputencoding` spécifie l'encodage ; `literate` définit les substitution de caractères. Toutes ces substitutions sont de la forme {caractère dans le listing}{ce par quoi on doit le replacer}{nombre d'espaces à droite et à gauche de la substitution}.

2.13 Objets flottants : figures et tableaux

La plupart des publications contiennent un nombre important de figures et de tableaux. Ces éléments nécessitent un traitement particulier car ils ne peuvent être coupés par un changement de page. On pourrait imaginer de commencer une nouvelle page chaque fois qu'une figure ou un tableau ne rentrerait pas dans la page en cours. Cette façon de faire laisserait de nombreuses pages à moitié blanches, ce qui ne serait réellement pas beau. La solution est de laisser «flotter» les figures et les tableaux qui ne rentrent pas sur la page en cours, vers une page suivante et de compléter la page avec le texte qui suit l'objet «flottant». LaTeX fournit deux environnements pour les *objets flottants* adaptés respectivement aux figures (`figure`) et aux tableaux (`table`). Pour faire le meilleur usage de ces deux environnements, il est important de comprendre

Caractère	Emplacement pour l'objet flottant...
`h`	*here*, ici, à l'emplacement dans le texte où la commande se trouve. Utile pour les petits objets.
`t`	*top*, en haut d'une page
`b`	*bottom*, en bas d'une page
`p`	*page*, sur une page à part ne contenant que des objets flottants.
`!`	ici, sans prendre en compte les paramètres internes (tels que le nombre maximum d'objets flottants sur une page) qui pourraient empêcher ce placement.

TABLE 2.1 – Placements possibles

comment LaTeX traite ces objets flottants de manière interne. Dans le cas contraire ces objets deviendront une cause de frustration intense car LaTeX ne les placera jamais à l'endroit où vous souhaitiez les voir.

Tout objet inclus dans un environnement `figure` ou `table` est traité comme un objet flottant. Les deux environnements flottants ont un paramètre optionnel (voir le tableau 2.1) : `\begin{figure}[placement]` ou `\begin{table}[placement]`. Ce paramètre permet de dire à LaTeX où vous préférez positionner l'objet à flotter si c'est possible. En général, si un objet ne peut pas être placé sur la page en cours, il est placé soit dans la file des figures soit dans la file des tableaux[6]. Quand une nouvelle page est entamée, LaTeX essaye d'abord de voir si les objets en tête des deux files pourraient être placés sur une page spéciale, à part. Si cela n'est pas possible, les objets en tête des deux files sont traités comme s'ils venaient d'être trouvés dans le texte : il essaye de les placer selon leurs spécifications de placement (sauf `h`, qui n'est plus possible). Tous les nouveaux objets flottants rencontrés dans la suite du texte sont ajoutés à la queue des files. LaTeX respecte scrupuleusement l'ordre d'apparition des objets flottants. C'est pourquoi un objet flottant qui ne peut être placé dans le texte repousse tous les autres à la fin du document. D'où la règle :

> «Si LaTeX ne place pas les objets flottants comme vous le souhaitez, c'est souvent à cause d'un seul objet trop grand qui bouche l'une des deux files d'objets flottants.»

Essayer d'imposer à LaTeX un emplacement particulier pose souvent problème : si l'objet flottant ne tient pas à l'emplacement demandé, alors il est coincé et bloque le reste des objets flottants. En particulier, l'utilisation de la seule option `[h]` pour un flottant est une idée *à proscrire*, les versions modernes de LaTeX changent d'ailleurs automatiquement l'option `[h]` en `[ht]`.

Voici quelques éléments supplémentaires qu'il est bon de connaître sur les environnements flottants.

- La commande `\caption` permet de définir une légende pour l'objet. Un numéro (incrémenté automatiquement) et le mot «Figure» ou «Table» (traduits automatiquement en français par le package `babel`) sont ajoutés par LaTeX.
- Avec `\label` et `\ref` vous pouvez faire référence à votre objet à l'intérieur de votre texte. La commande `label` doit apparaître *après* la commande `\caption` d'une légende si vous voulez référencer le numéro de cette légende.
- Les deux commandes `\listoffigures` et `\listoftables` fonctionnent de la même manière que la commande `\tableofcontents` : elles impriment respectivement la liste des figures et des tableaux. Dans ces listes, la légende est reprise en entier. Si vous désirez utiliser des légendes longues, vous pouvez en donner une version courte entre crochets qui sera utilisée pour la table : `\caption[courte]{Longue sous la figure}`

2.13.1 Insérer des images

L'importation d'une image dans un document se fait en deux étapes : d'abord on indique à LaTeX le nom du fichier qui contient l'image, puis on la transforme en un objet flottant (appelé `figure`) pour que LaTeX puisse la placer dans la page avec le meilleur rendu possible. Cela signifie que, si l'image est trop grande

[6] Il s'agit de files FIFO (*First In, First Out*) : premier arrivé, premier servi.

pour le peu de place qui reste dans la page, il la placera sur la page d'après. Il ne faut pas forcer le placement des images car LaTeX est plus performant que vous pour l'arrangement des pages, donc il faut le laisser faire son travail.

Insérer une image : les formats d'images acceptés par une compilation avec `pdflatex` sont JPG, PNG et `pdf` (le dernier est à préférer pour tout ce qui est vectoriel). Pour inclure une image appelée `courbe.png`, présente dans le même répertoire que le fichier `tex` que vous compilez (très important), il faut charger le package `graphicx` en mettant dans le préambule

```
\usepackage{graphicx}
```

puis mettre dans le corps du document là où on veut faire apparaître l'image la commande

```
\includegraphics{courbe}
```

Noter qu'il n'y a pas besoin de spécifier l'extension.[7]

La commande `\includegraphics` peut prendre un certain nombre d'options. L'option `width` permet de spécifier la largeur de l'image. Par exemple

```
\includegraphics[width=0.5\textwidth]{courbe}
```

mettra la largeur de l'image à la moitié de la largeur du texte, la hauteur étant changée en conséquence pour garder les proportions de l'image. De même, on peut spécifier la hauteur

```
\includegraphics[height=2cm]{courbe}
```

On peut aussi changer la taille de l'image avec l'option `scale`. Par exemple

```
\includegraphics[scale=0.4]{courbe}
```

réduira la taille de l'image à 40% de celle de l'original. On peut changer l'orientation de l'image avec l'option `angle`. Par exemple

```
\includegraphics[angle=90]{courbe}
```

tourne l'image de 90 degrés (sens trigonométrique). On peut couper les bords de l'image avec les options `trim` et `clip`. Par exemple

```
\includegraphics[trim=1cm 2cm 3cm 4cm,clip]{courbe}
```

affichera l'image coupée de 1 cm à partir de la marge gauche, 2 cm de la marge en bas, 3 cm de la marge droite et 4 cm de la marge haute.

Pour une description des autres options, voir la documentation du package `graphicx` disponible à l'adresse https://www.ctan.org/pkg/grfguide

Exercice 4

Placer une image dans la marge comme ci-contre.

Placer l'image dans une figure flottante : il est conseillé de laisser LaTeX s'occuper automatiquement du placement d'une image grâce à l'environnement `{figure}`. À l'intérieur de cet environnement, il faut mettre la commande `\includegraphics` et rajouter une légende afin de pouvoir faire référence à la figure. Pour cela, on utilise la commande `\caption` ainsi que la commande `\label` pour y faire référence plus tard avec la commande `\ref` (comme pour les numéros de sections, d'équations, etc.). La syntaxe est donc

```
\begin{figure}
\centering
\includegraphics{myfig}
\caption{Titre de la figure}\label{fig.myfig}
\end{figure}
```

[7] Recommandation : remplacer le package `graphicx` par le nouveau package `graphbox` : ce package charge `graphicx` en sous-main et ajoute des options pour l'alignement (utile pour aligner des images horizontalement par exemple).

Toujours mettre le `\label` après le `\caption`. On peut encourager LaTeX à positionner l'image à un certain endroit en utilisant un argument optionnel de l'environnement `{figure}` (par exemple `\begin{figure}[t]`). Le paramètre `t` favorisera l'apparition en haut de page, le paramètre `b` en bas de page, le paramètre `h` à l'endroit où est `{figure}` dans le fichier source et `p` sur une page à part avec d'autres images.

La figure 2.2 a été obtenue par le code suivant :

```
\lipsum[1]
On voit à la figure~\ref{fig.hippopotenuse} que bla bla

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{hippopotenuse}
\caption{I put the hippo in hippopotenuse}\label{fig.hippopotenuse}
\end{figure}

\lipsum[2]
```

Remarquer que dans le code on insère l'image après le premier paragraphe mais LaTeX a préféré la faire apparaître en haut de la page. Cela n'est pas gênant car il suffit de faire référence à l'image non pas par une expression du type «comme on voit sur la FIGURE ci-dessous» mais plutôt par une expression du type «comme on voit sur la FIGURE 1» où le numéro est géré automatiquement par un mécanisme d'étiquette/référence.

> Le comportement par défaut des images est de flotter, afin que LaTeX puisse trouver la meilleure façon de les disposer dans votre document et d'en améliorer l'apparence. Si vous jetez un coup d'oeil, vous verrez que c'est ainsi que les livres sont souvent mis en page. La meilleure chose à faire est donc de laisser LaTeX faire son travail et de ne pas essayer de forcer le placement des figures à des endroits spécifiques. Cela signifie également que vous devez éviter d'utiliser des phrases telles que « `dans la figure suivante` » qui exige que la figure soit placée à un endroit précis, et utiliser plutôt « `dans la figure~\ref{...}` », en tirant parti des références croisées de LaTeX .
>
> Dans certains cas il peut s'avérer nécessaire d'utiliser la commande `\FloatBarrier` du package placeins : elle ordonne à LaTeX de placer tous les objets flottants avant cette commande (par exemple, avant de commencer une nouvelle section). Ça reste un objet flottant mais on empêche LaTeX de le placer au delà de cette commande.
>
> Si, pour une raison quelconque, vous voulez vraiment qu'une figure particulière soit placée "ICI", et non là où LaTeX veut la placer, vous pouvez utiliser l'option `[H]` du package `float` qui transforme la figure flottante en une figure non flottante normale. Notez cependant que dans ce cas vous ne voulez pas forcement ajouter de légende à votre figure, vous n'avez alors pas besoin d'utiliser l'environnement `figure` du tout! Vous pouvez utiliser juste la commande `\includegraphics` n'importe où dans votre document pour insérer une image.

2.13.2 Insérer des tableaux

Le rendu d'un tableau dans un document se fait aussi en deux étapes : d'abord on construit le tableau, puis on le transforme en un objet flottant (appelé `table`) pour que LaTeX puisse le placer dans la page avec le meilleur rendu possible, selon le même mécanisme déjà vu pour les images/figures.

Construire un tableau : pour écrire un tableau, on utilise l'environnement `{tabular}`. Cet environnement prend un argument obligatoire qui est la spécification du type des colonnes. Les types de colonnes de base sont `c` (centré), `l` (aligné à gauche), `r` (aligné à droite) et `p{5cm}` (paragraphe de 5cm de large). Par exemple, pour obtenir un tableau avec trois colonnes, la première étant centrée, la deuxième un paragraphe de 7 cm et la dernière alignée à droite, on écrit

```
\begin{tabular}{cp{7cm}r}
...
\end{tabular}
```

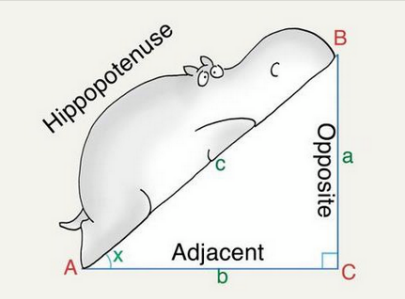


FIGURE 1 – I put the hippo in hippopotenuse

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer 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. On voit à la figure 1 que bla bla

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.

FIGURE 2.2 – Exemple de `figure` avec référence.

Le texte des différentes colonnes est séparé par un & et on passe à la ligne suivante en utilisant \\. Par exemple :

```
Voici un tableau:

\begin{tabular}{cp{1cm}r}
titi toto       & tutu tata & tete  \\
titi toto       & tutu tata & tete
\end{tabular}
```

Voici un tableau :

titi toto	tutu tata	tete
titi toto	tutu tata	tete

La commande \\ doit uniquement être utilisée pour passer à la ligne dans les tableaux. Elle ne doit pas être utilisée pour aller à la ligne lorsqu'on tape du texte (il faut laisser une ligne blanche dans le fichier source). L'utilisation abusive de \\ dans un document que vous me rendrez sera sévèrement sanctionné. (Noter que le tableau est la suite du texte ; pour éviter ceci, le mettre dans un environnement `{center}` ou le rendre un flottant).

Pour rajouter un trait vertical entre deux colonnes, il suffit de mettre un | (qu'on obtient avec la combinaison de touches «AltGr+6») dans la spécification des colonnes. Pour tracer un trait horizontal, il y a la commande `\hline`. Cette commande doit être soit au tout début du tableau soit après un \\. Voici un exemple :

```
\begin{center}
\begin{tabular}{|c|p{1cm}r|}
\hline
bla bla & bla bla bla & bla      \\
\hline
bla     & bla bla     & bla bla  \\
\hline
\end{tabular}
\end{center}
```

bla bla	bla bla bla	bla
bla	bla bla	bla bla

Lorsqu'on fait des tableaux, toujours charger le package `array` (dans le préambule et, comme toujours, avant `hyperref`), qui corrige certains petits problèmes concernant les raccords entre traits horizontaux et verticaux et étend les possibilités pour les tableaux.

LaTeX traite le contenu d'un environnement `tabular` comme une boîte indivisible, en particulier il ne peut y avoir de coupure de page. Pour réaliser de longs tableaux s'étendant sur plusieurs pages il faut avoir recours aux extensions supertabular ou longtable.

Parfois les tableaux par défaut de LaTeX donnent une impression d'étroitesse. Si vous voulez leur donner plus d'extension, vous pouvez le faire en modifiant les valeurs de `arraystretch` et `tabcolsep` comme dans l'exemple suivant.

```
{
\renewcommand{\arraystretch}{1.5}
\renewcommand{\tabcolsep}{0.5cm}
\begin{center}
\begin{tabular}{|c|p{1cm}r|}
\hline
bla bla & bla bla bla & bla      \\
\hline
bla     & bla bla     & bla bla  \\
\hline
\end{tabular}
\end{center}
}
```

bla bla	bla bla bla	bla
bla	bla bla	bla bla

Pour aller plus loin, voir le package booktabs.

Pour fusionner des cellules d'une même ligne on utilisera `\multicolumn{nombre de cellules à fusionner}{formatage}{contenu}` :

```
\begin{center}
\begin{tabular}{|clr|}
\hline
bla bla & bla bla bla & bla      \\
\hline
\multicolumn{2}{|c}{fusionnées}  & bla bla  \\
\hline
bla bla & bla bla bla & bla      \\
\hline
\end{tabular}
\end{center}
```

bla bla	bla bla bla	bla
fusionnées		bla bla
bla bla	bla bla bla	bla

Exercice 5

Reproduire le tableau suivant.

AA	BBB	C
D	EE	FFF
GGG	H	II

Placer le tableau dans une `table` flottante : comme pour les images, il est conseillé de laisser LaTeX s'occuper automatiquement du placement d'un tableau grâce à l'environnement `{table}`. À l'intérieur de cet environnement, il faut mettre l'environnement `{tabular}` et rajouter une légende afin de pouvoir faire référence au tableau. La syntaxe est donc

```
\begin{table}
\centering
\begin{tabular}{...}
...
\end{tabular}
\caption{Titre du tableau}\label{tab.montableau}
\end{figure}
```

Tout ce qu'on a dit pour l'environnement `{figure}` reste valable pour l'environnement `{tabular}`.

2.14 Mettre son document sur plusieurs colonnes

Si on souhaite que tout le document soit écrit sur deux colonnes, on peut utiliser l'option de classe

```
\documentclass[...,twocolumn]{...}
```

2.14.1 `multicol(s)`

Parfois on peut être conduit à écrire seulement une partie du document sur plusieurs colonnes, par exemple si on veut mettre côte à côte un texte et une image. La méthode la plus simple consiste à utiliser l'environnement `multicols` qui a besoin d'un paramètre : le nombre de colonnes que l'on veut.

Dans le préambule du document, il faut importer le paquet `multicol` (sans s)

```
\usepackage{multicol}
```

Puis, dans le corps du document du document, il faudra utiliser l'environnement `multicols` (avec un s) :

```
\begin{multicols}{nb colonnes}
...
\end{multicols}
```

Le nombre de colonnes est compris entre 1 et 10.

Voici un exemple :

```
\documentclass[a4paper,10pt]{scrartcl}

\usepackage{multicol}
\usepackage{lipsum}

\begin{document}
\lipsum[1]
\begin{multicols}{2}
\lipsum[2]
\end{multicols}
\lipsum[3]
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer 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.

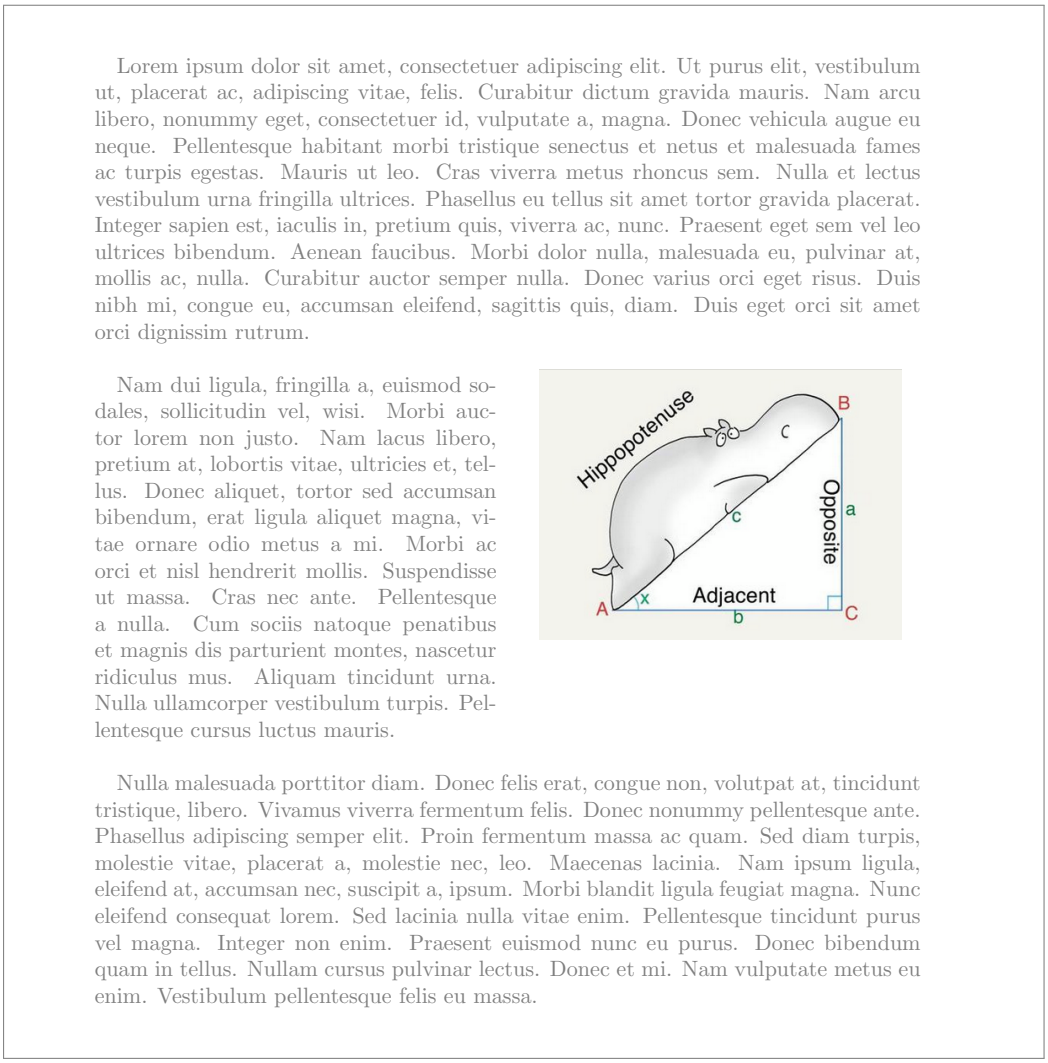
Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit 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.

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.

Le texte est réparti automatiquement sur les deux colonnes. On peut cependant forcer LaTeX à effectuer un changement de colonne avec l'instruction `\columnbreak` suivie d'une ligne vide.

```
\documentclass[a4paper,10pt]{scrartcl}

\usepackage{multicol}
\usepackage{lipsum}
\usepackage{graphicx}

\begin{document}
\lipsum[1]
\begin{multicols}{2}
\lipsum[2]
```

```
\columnbreak

\centering
\includegraphics[width=0.9\columnwidth]{hippopotenuse}
\end{multicols}
\lipsum[3]
\end{document}
```



Avec `multicols` les colonnes ont toutes la même largeur, et ce n'est pas forcément ce que l'on veut : on pourrait par exemple vouloir réduire la largeur de la colonne contenant l'image, et donc élargir l'autre. Il y a, bien sûr, une solution : l'environnement `minipage`.

2.14.2 `minipage`

Pour créer une boîte qui simule la création d'une page avec d'éventuelles notes de bas de page, tableaux, listes, etc. on peut utiliser l'environnement `minipage`

```
\begin{minipage}[pos]{largeur}
```

```
...
\end{minipage}
```

Le paramètre pos contrôle l'alignement vertical de la boîte par rapport à la base du texte précédent. Trois possibilités : la lettre c (par défaut, positionne la minipage de sorte que son centre vertical soit aligné avec le centre des lignes de texte adjacentes), la lettre t (aligne la ligne du haut de la minipage sur la ligne de base du texte l'entourant) ou la lettre b (aligne la ligne du bas de minipage avec la ligne de base du texte l'entourant).

Voici un exemple :

```
Une petite ligne avant la minipage. Dans cet exemple on a créé une minipage faisant 30\% de
 la largeur du texte et contenant un environnement \texttt{itemize}.
\medskip % espace verticale

\begin{minipage}[pos=b]{0.3\textwidth}
Ma mini-page très étroite qui contient une liste à puce:
\begin{itemize}
\item Bonjour\footnote{Une note}
\item Au-revoir
\end{itemize}
\end{minipage}
\hspace{2cm} % espace horizontale
\begin{minipage}[pos=c]{0.3\textwidth}
Et une deuxième mini-page juste à coté pour voir l'alignement.
\end{minipage}
\medskip % espace verticale

Encore du texte. Les paragraphes avant et après la boîte sont un peu trop collés à la
minipage, il faut alors ajouter de l'espace verticalement.
```

Une petite ligne avant la minipage. Dans cet exemple on a créé une minipage faisant 30% de la largeur du texte et contenant un environnement itemize.

Ma mini-page très étroite qui contient une liste à puce :

— Bonjour

— Au-revoir

Et une deuxième mini-page juste à coté pour voir l'alignement.

Encore du texte. Les paragraphes avant et après la boîte sont un peu trop collés à la minipage, il faut alors ajouter de l'espace verticalement.

Avec une minipage on a besoin d'introduire des espacements verticaux. Voici ceux disponibles :

```
\vspace{<longueur>} % espacement vertical
\vspace*{<longueur>} % espacement vertical incompressible
\smallskip % idem que \vspace{\smallskipamount}
\medskip % idem que \vspace{\medskipamount}
\bigskip % idem que \vspace{\bigskipamount}
\\ % saut d'une ligne
\hspace{<longueur>} % espacement horizontal
\hspace*{<longueur>} % espacement horizontal incompressible
```

2.15 Le formatage du texte ou comment écrire en *italique* en gros en petit en `chasse fixe...`

Dans cette section on va apprendre à modifier les caractéristiques d'une police, c'est à dire à changer de fonte. Avec ces commandes on va pouvoir personnaliser un peu plus les documents. Attention toutefois de ne pas en faire de trop, on risquerait de rendre le document si ce n'est illisible tout du moins très moche.

La police utilisée dans ce polycopié est `fourier`, mais vous pouvez utiliser `lmodern` en chargeant le package `lmodern`. Une police possède 4 caractéristiques que l'on peut faire varier individuellement : la famille, la graisse, la forme, la taille (le corps). On peut donc avoir de nombreuses combinaisons possibles comme par exemple gras et italique. Toutefois toutes les combinaisons ne sont pas possibles et dépendent de la police utilisée. Par exemple en `lmodern` on ne peut pas obtenir des petites capitales en italique, ni en fonte sans serif.

La police c'est l'ensemble des caractères de toutes tailles, graisses, forme, d'une même famille. Une fonte est un ensemble de caractères d'une même police ayant la même graisse, la même taille, la même forme. Par exemple la police de ce paragraphe est `fourier`, la fonte utilisée est serif corps 10pt (`fourier`, sans serif, corps 8pt, italique est une autre fonte de la même police.)

Il existe deux types de commandes pour changer de fonte, les commandes à arguments et les commandes déclaratives :

- Les commandes à arguments servent pour modifier un mot ou un petit groupe de mots qui ne dépasse pas la taille d'un paragraphe. Un saut de paragraphe à l'intérieur d'une commande à argument déclenche une erreur de compilation. Elles commencent toutes par `\text` et prennent l'argument entre accolades. Par exemple la commande à argument pour mettre en gras est `\textbf{}` :

```
\textbf{gras}
```

gras

- Les commandes déclaratives agissent à partir de l'endroit ou elles sont positionnées dans le texte sur toute la suite du texte jusqu'à ordre contraire. Un ordre contraire peut être donné par une autre commande déclarative annulant la première. Par exemple la commande graisse normale annulant la commande gras. La portée d'une commande déclarative peut être limitée par des accolades. `\bfseries` est la commande déclarative pour mettre en gras :

```
{\bfseries Tout ce qui se trouve ici même
 après un paragraphe ou des sections
\section*{Une section en gras}
sera en gras jusqu'à cette accolade} et
ici tout revient normal
```

Tout ce qui se trouve ici même après un paragraphe ou des sections

Une section en gras

sera en gras jusqu'à cette accolade et ici tout revient normal

Une autre façon de limiter la portée d'une commande déclarative est de l'utiliser dans un environnement. Un environnement commence par un `\begin{environnement}` et se termine par un `\end{environnement}`

```
\begin{center}
\bfseries Tout ce qui se trouve dans cet
environnement sera centré et en gras.
\end{center}
Après la fermeture de l'environnement on
retrouve une graisse normale.
```

Tout ce qui se trouve dans cet environnement sera centré et en gras.

Après la fermeture de l'environnement on retrouve une graisse normale.

On peut mettre plusieurs commandes déclaratives dans le même environnement ou la même paire d'accolades.

```
{\Large \rmfamily \itshape En gros, roman
 et italique}
```

En gros, roman et italique

Les familles de police

Nom	Commande à argument	Commande déclarative	Résultat
Roman	`\textrm{Roman}`	`{\rmfamily Roman}`	Roman
Sans serif	`\textsf{Empattement}`	`{\sffamily Empattement}`	Empattement
Chasse fixe	`\texttt{typewriter}`	`{\ttfamily typewriter}`	`typewriter`

La graisse

Nom	Commande à argument	Commande déclarative	Résultat
Graisse normale	`\textmd{Normale}`	`{\mdseries Normale}`	Normale
Écrit en gras	`\textbf{En gras}`	`{\bfseries En gras}`	**En gras**

La forme

Nom	Commande à argument	Commande déclarative	Droite
Écriture droite	`\textup{Droite}`	`{\upshape Droite}`	Normale
En italique	`\textit{Italique}`	`{\itshape Italique}`	*Italique*
Forme penchée	`\textsf{Penchée}`	`{\slshape Penchée}`	Penchée
En petites capitales	`\textsc{Petites capitales}`	`{\scshape Petites capitales}`	Petites capitales
En emphase	`\emph{Emphatisé}`	`{\em Emphatisé}`	*Emphatisé*

Avec LaTeX on ne peut pas modifier localement la taille du texte en augmentant le nombre de points (pt) comme on peut faire avec les traitements de texte WYSIWYG. On doit se contenter de taille relative par rapport à la taille normale du texte. Celle-ci est passée comme option à la commande `\documentclass`. LaTeX accepte 3 tailles différentes : 10pt, 11pt, 12pt. Avec KOMA-Script on peut utiliser d'autres valeurs (14pt par exemple) mais on peut rencontrer des problèmes de compatibilité avec les autres packages. Notons que, de plus, le changement de taille se fait uniquement avec une commande déclarative :

```
{\tiny Texte}
{\scriptsize Texte}
{\footnotesize Texte}
{\small Texte}
{\normalsize Texte}
{\large Texte}
{\Large Texte}
{\huge Texte}
{\Huge Texte}
```

Texte Texte Texte Texte Texte Texte Texte Texte Texte

Dans un manuscrit réalisé sur une machine à écrire, les mots importants sont valorisés en les soulignant ; on peut obtenir ce résultat en LaTeX avec la commande `\underline` :

```
... valorisés en les \underline{soulignant};
 on peut obtenir ...
```

... valorisés en les soulignant ; on peut obtenir ...

Toutefois, dans un ouvrage imprimé on préfère les *mettre en valeur*[8]. Les commandes de mise en valeur sont `\emph` et `\em`. La plupart du temps la mise en évidence du mot ou du groupe de mot se fera en le mettant en italique. Mais dans le cas d'un texte en italique `\emph{}` et `\em` mettront le texte en écriture droite.

```
Voici un texte \emph{normal} avec le mot
\emph{normal} en emphase.

{\itshape Voici un texte en \emph{italique}
 avec le mot \emph{italique} en emphase.}
```

Voici un texte *normal* avec le mot *normal* en emphase.
Voici un texte en italique *avec le mot* italique *en emphase.*

[8] *Emphasize* en anglais.

2.16 Définition d'environnements et de commandes personnelles

Il est possible de définir des commandes et des environnements personnels.

- **Exemple de définition d'une commande personnelle :** supposons de vouloir mettre les noms de famille en petites capitales, pour cela on doit utiliser la commande `\textsc` qui prend un argument et le met en petites capitales. Voici un exemple :

```
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}.
```

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.

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 composent 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. Notons que les espaces après une commande sont ignorées par LaTeX . Si on veut obtenir une espace, on doit terminer la commande par des accolades et laisser une espace comme dans cet exemple :

```
bla \SNCF bla

bla \SNCF{} bla
```

bla S.N.C.Fbla
bla S.N.C.F bla

Ce serait une très mauvaise idée de mettre une espace dans la définition de la commande, car on aurait toujours une espace, y compris avant une ponctuation. Pour gérer ces deux situations on peut utiliser le package `xspace` :

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

```
bla \SNCF bla

bla \SNCF.
```

bla S.N.C.F bla
bla S.N.C.F.

La commande teste ce qui suit la commande : si c'est une ponctuation ou { ou }, elle ne fera rien ; dans les autres cas, elle ajoute une espace. Une conséquence de ce fonctionnement est qu'une `\footnote` suivant `\SNCF` va produire une espace inopportune. Elle peut être évitée en tapant {} : `bla \SNCF{}\footnote{Toto}`.

- 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 6

Créer une commande à 2 arguments \nompropre qui imprime le prénom suivi du nom en petites capitales et l'utiliser pour reproduire l'exemple initiale de la section.

Exercice 7

Modifier la commande précédente pour qu'elle n'affiche plus que le nom de famille.

Exercice 8

Modifier la commande précédente pour qu'elle affiche le nom de famille en couleur.

- **Exemple de définition d'un environnement personnel :** on veut écrire un environnement {citationFR} pour mettre en page une citation comme suit : le texte cité se trouve dans un environnement {quotation} entouré par des guillemets. Pour définir notre environnement on utilise la syntaxe suivante :

```
\newenvironment{citationFR}{\begin{quotation}\og}{\fg\end{quotation}}
```

qui crée un environnement {citationFR} sans arguments tel que

```
\begin{citationFR}
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
\end{citationFR}
```

produit

« bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla »

2.17 Pour aller plus loin

Voici quelques packages que vous pouvez utiliser pour personnaliser vos documents (avec parcimonie) :

— la documentclass KOMA-Script :
  — Guide complète https://komascript.de/~mkohm/scrguien.pdf
  — Traduction et adaptation du guide https://archives.framabook.org/koma-script/index.html
— gestion des marges :
  — typearea (package automatiquement chargé par les classes koma-script comme scrartcl) https://www.ctan.org/pkg/typearea.html
  — geometry https://www.ctan.org/pkg/geometry.html
— gestion des entêtes et pieds de page :
  — scrlayer-scrpage voir le chapitre 5 de la documentation de KOMA-Script
  — fancyhdr https://www.ctan.org/pkg/fancyhdr.html
— tableaux complexes :

— `array` https://www.ctan.org/pkg/array.html
— `booktabs` https://www.ctan.org/pkg/booktabs.html
— `longtable` https://www.ctan.org/pkg/longtable.html
— `tabularx` https://www.ctan.org/pkg/tabularx.html
— `tcolorbox` https://www.ctan.org/pkg/tcolorbox
— `tabu` https://ctan.org/pkg/tabu
— Quelque conseils : https://people.inf.ethz.ch/markusp/teaching/guides/guide-tables.pdf

3 Mathématiques

LaTeX est aujourd'hui le standard utilisé dans l'édition mathématique (que ce soit pour des articles de recherche, des livres, des polycopiés de cours, des feuilles de TD, etc.). Le but de ce chapitre est d'apprendre à taper des formules simples avec LaTeX . Au début, cela fait beaucoup de commandes à mémoriser, mais avec la pratique, on finit par connaître la plupart des commandes et cela permet de taper les formules relativement rapidement. Ne pas hésiter à utiliser les menus de symboles de `Texmaker` dans le panneau de gauche.

3.1 Documents avec théorèmes, propositions, etc.

Les environnements pour écrire des théorèmes, des corollaires, des lemmes et autres propositions ainsi que les démonstrations, les exercices etc. ne sont pas définis par défaut.

Pour apprendre à produire un document qui contient ces environnements, on va analyser l'exemple donné à la figure 3.1. On remarque que 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.

On note d'ailleurs que la numérotation est automatique et choisie de telle façon que le numéro de section est 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` dans le préambule. 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 :

```
\usepackage{amsthm}
```

- On doit maintenant définir un environnement qui permettra d'écrire le théorème proprement dit. Dans le préambule on écrit

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

  Le premier argument est le nom de l'environnement [1] ; il ne doit pas comporter d'accents. Si on utilise `theoreme`, on tapera plus tard dans le corps du document `\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 `scrbook` au lieu de `scrartcl`, on voudrait probablement mettre `chapter` ici au lieu de `section`.

[1] 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 þ).

1 Rappels

Définition 1.1. Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero.

Proposition 1.2. *Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.*

Démonstration. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

Notons le petit carré automatiquement inséré pour signaler la fin de la preuve (il s'appelle QED, acronyme de "quod erat demonstrandum"). □

Corollaire 1.3. *Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.*

Exercice 1. *Mauris ut leo. Cras viverra metus rhoncus sem.*

Exercice 2. *Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat.*

2 Approfondissements

Définition 2.1. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

Lemme 2.2. *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.*

Démonstration. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. □

Théorème 2.3. *Cras nec ante. Pellentesque a nulla.*

Démonstration. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. □

Remarque. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Exercice 3. *Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.*

FIGURE 3.1 – Exemple d'un document avec théorèmes, propositions, etc.

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

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

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

```
\newtheorem{exo}{Exercice}
```

- Il faut enfin sélectionner un style ; cela se fait avec la commande `\theoremstyle`. Le style `plain` compose le nom de l'environnement en gras et son contenu en italique ; le style `definition` compose le nom de l'environnement en gras et son contenu en droit ; le style `remark` compose le nom de l'environnement en italique et son contenu en droit.

```
\theoremstyle{plain}
    \newtheorem{theoreme}{Théorème}[section]
    \newtheorem{proposition}[theoreme]{Proposition}
    \newtheorem{corollaire}[theoreme]{Corollaire}
    \newtheorem{lemme}[theoreme]{Lemme}
    \newtheorem{exo}{Exercice}
\theoremstyle{definition}
    \newtheorem{definition}[theoreme]{Définition}
\theoremstyle{remark}
    \newtheorem*{remarque}{Remarque}
```

Il n'y a pas de styles pour les démonstrations, elles sont produites directement en utilisant dans le corps du document l'environnement `{proof}` :

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

Pour personnaliser l'apparence de ces environnements on pourra s'appuyer sur l'un des package suivants :

— `ntheorem` https://ctan.org/pkg/ntheorem

— `thmtools` https://ctan.org/pkg/thmtools

— `bclogo` https://ctan.org/pkg/bclogo

Exercice 9

En utilisant les outils qu'on vient de décrire, reproduire le document de la figure 3.1.

3.2 Les différents modes mathématiques

Pour afficher des formules mathématiques avec LaTeX, il faut lui indiquer qu'il s'agit d'un environnement mathématiques. Il existe deux modes mathématiques :

① les formules dans le texte (appelée *textstyle* ou *inline*),

② les formules en évidence (dites *displaystyle*). Ces dernières peuvent être sans ou avec numérotation.

- **Formule dans le texte**

  Pour mettre une formule dans le texte, il suffit de mettre la formule entre deux dollars :

```
Soit $y=f(x)$ alors
```

  Soit $y = f(x)$ alors

- **Formule en évidence, sans numérotation**

  Pour mettre une formule en évidence, il faut mettre la formule entre `\[` et `\]` :

```
Soit
\[
y=f(x)
\]
alors
```

  Soit

$$y = f(x)$$

  alors

  On peut aussi utiliser l'environnement `{equation*}` :

```
Soit
\begin{equation*}
y=f(x)
\end{equation*}
alors
```

  Soit

$$y = f(x)$$

  alors

  Attention : laisser des lignes blanches en mode displaystyle produit une erreur de compilation.

  Nota Bene : l'utilisation des doubles dollars `$$...$$` (qu'on peut trouver dans certains vieux manuels et encore trop souvent sur internet) pour mettre une formule en évidence sera considéré comme une faute grave et sévèrement sanctionné. [2]

- **Formule en évidence, avec numérotation**

  Pour une équation numérotée, il faut utiliser l'environnement `{equation}` :

```
Soit
\begin{equation}
y=f(x)
\end{equation}
alors
```

  Soit

$$y = f(x) \tag{3.1}$$

  alors

  Pour faire référence à une équation numérotée, il faut placer un `\label` avec une étiquette, par exemple :

```
Soit
\begin{equation}\label{eq.fonction.f}
y=f(x)
\end{equation}
alors
```

  Soit

$$y = f(x) \tag{3.2}$$

  alors

  puis utiliser `\eqref{eq.fonction.f}` pour imprimer (3.2), `\pageref{eq.fonction.f}` pour imprimer le numéro de page (ici 42) où se trouve l'équation (on peut aussi utiliser `\ref{eq.fonction.f}` pour imprimer 3.2, c'est-à-dire le numéro de l'équation sans les parenthèses). Comme la table des matières, ces commandes requièrent **deux compilations** successives pour fonctionner correctement. Se rappeler que les commandes `\label`, `\ref` et `\pageref` ne sont pas limitées aux équations, mais fonctionnent aussi pour des sections ou des théorèmes (en fait, pour tout ce qui est numérotés automatiquement).

Exercice 10

Reproduire le document de la figure 3.2.

[2] Voir par exemple le document "Liste des péchés des utilisateurs de LaTeX " pour plus de détails https://ctan.org/tex-archive/info/l2tabu/french

Soit f une fonction vérifiant

$$f(x) = 2x + 1. \tag{1}$$

On a $f(x) - 1 = 2x$ d'après la formule (1).

FIGURE 3.2 – Exemple de document avec deux simples formules mathématiques.

Propriétés des *modes mathématiques* :

— les espaces saisis au clavier sont ignorés ;

— tous les caractères alphabétiques sont en italique mais différentes de l'italique en *mode texte*

```
y = f(x) est différentiable

$y = f(x) est différentiable$

$y = f(x)$ est différentiable
```

y = f(x) est différentiable
$y = f(x) est différentiable$
$y = f(x)$ est différentiable

Nota Bene : la non utilisation du mode mathématique pour une formule (resp. l'utilisation du mode mathématique pour du texte) sera considéré comme une faute grave et sévèrement sanctionné.

On verra plus loin comment ajouter des espaces ou écrire du texte dans un environnement mathématique.

3.3 Indices et exposantes

Deux opérations fondamentales en mode mathématique sont la mise en exposant et la mise en indice.

- Pour obtenir un indice, il faut utiliser la touche _ qui s'obtient avec la touche «8». Par exemple, taper `$x_1$` donnera x_1. Attention, taper `$x_12$` ne donne pas x_{12} mais $x_1 2$: seul le premier caractère tapé après _ est mis en indice ; pour obtenir x_{12}, il faut taper `$x_{12}$`.
- Pour obtenir un exposant, il faut utiliser ^ (accent circonflexe qui s'obtient en tapant deux fois la touche «^»). Par exemple, `$x^2$` donne x^2. De même que pour les indices, `$x^23$` donne $x^2 3$ tandis que `$x^{23}$` donne x^{23}.

On peut bien sûr combiner les deux, dans l'ordre que l'on veux : `$x_1^2$` ou `$x^2_1$` donnent x_1^2.

Exercice 11

Taper les formules suivantes :

$$(x^2)^3 = x^{2^3}$$

$$F_n = 7^{2^n} + 1$$

Remarquer la différence dans les formules suivantes :

```
$(x^2)^3$  et  ${(x^2)}^3$  et  $\left(x^2\right)^3$  et  ${\left(x^2\right)}^3$
```

$(x^2)^3$ et ${(x^2)}^3$ et $\left(x^2\right)^3$ et ${\left(x^2\right)}^3$

3.4 Racine carrée, racine n-ième

La racine carrée s'obtient par `\sqrt{...}` et la racine n-ième par `\sqrt[n]{...}`.

```
\[
\sqrt{1+x} + \sqrt[3]{1+x}
\]
```

$$\sqrt{1+x}+\sqrt[3]{1+x}$$

Exercice 12

Taper les formules suivantes :

$$u_{n+1} = \sqrt[n]{1+u_n}$$

$$x_5 = \sqrt{1+\sqrt{2+\sqrt{3+\sqrt{4+\sqrt{5}}}}}$$

3.5 Texte dans une formule *displaystyle*

La commande `\text{}` permet d'insérer du texte dans une formule *displaystyle* :

```
\[
y = x^2 \text{ et donc } x = \pm\sqrt{y}
\]
```

$$y = x^2 \text{ et donc } x = \pm\sqrt{y}$$

Exercice 13

Taper la formule :

$$(\sqrt{x})^2 = x \text{ mais } \sqrt{x^2} \neq x \text{ en général.}$$

☞ Texte en indice

Il est fréquent que du texte soit utilisé en indice, par exemple $f_{\text{opt}}(x)$. Si les indices ou les exposants ont un rôle descriptif, ils sont à saisir en tant que texte et donc en argument de la commande `\text` :

```
Ne pas écrire $f_{opt}(x)$ mais $f_{\text{opt}}(x)$

Ne pas écrire $f_{\text{i}}(x)$ mais $f_i(x)$
```

Ne pas écrire $f_{opt}(x)$ mais $f_{\text{opt}}(x)$
Ne pas écrire $f_{\text{i}}(x)$ mais $f_i(x)$

3.6 Espaces en mode mathématique

Les espaces s'obtiennent via des commandes :

- 2 cadratins avec la commande `\qquad`

```
\[a \qquad b\]
```

$$a \qquad b$$

- 1 cadratin avec la commande `\quad`

```
\[a \quad b\]
```

$$a \quad b$$

- inter-mot avec la commande `\`

```
\[a \ b\]
```

$$a\ b$$

- épaisse avec la commande \;

```
\[a \; b\]
```

$$a \; b$$

- moyenne avec la commande \:

```
\[a \: b\]
```

$$a \: b$$

- fine avec la commande \,

```
\[a \, b\]
```

$$a \, b$$

- normale

```
\[a  b\]
```

$$a b$$

- fine négative avec la commande \!

```
\[a \! b\]
```

$$a \! b$$

3.7 Symboles d'usage courant

Certains symboles s'obtiennent directement au clavier :

```
$ ( ) [ ] | = + - / < > , ; : ! $
```

$$() [] | = + - / <>, ; : !$$

La plupart des symboles s'obtiennent via des commandes. [3]

Commande	\infty	\forall	\exists	\nexists	\partial	\pm	\mp
Résultat	∞	$\forall$	$\exists$	$\nexists$	∂	$\pm$	$\mp$
Commande	\times	\neq	\leq	\geq	\approx	\simeq	\equiv
Résultat	$\times$	$\neq$	$\leq$	$\geq$	$\approx$	$\simeq$	$\equiv$
Commande	\in	\subset	\cup	\cap	\setminus	\emptyset	\ell
Résultat	$\in$	$\subset$	$\cup$	$\cap$	$\setminus$	$\emptyset$	ℓ
Commande	\to	\mapsto	\implies	\impliedby	\iff	\nearrow	\searrow
Résultat	$\to$	$\mapsto$	$\implies$	$\impliedby$	$\iff$	$\nearrow$	$\searrow$

Pour la négation d'un symbole, on peut utiliser \not. Par exemple, `$F \not\subset E$` fournit $F \not\subset E$.

Exercice 14

Taper les formules suivantes :

$$y = x^2 \iff x = y^{1/2}$$

$$x > 0 \implies x^2 \neq 0$$

$$x \in X \setminus Y \implies x \notin Y$$

[3] L'application Web detexify permet de dessiner un symbole à la souris et obtenir la commande LaTeX correspondante.

3.8 Points de suspension

Commande	\cdot	\dots	\cdots	\ddots	\vdots
Résultat	$\cdot$	$\dots$	$\cdots$	$\ddots$	$\vdots$

La différence entre . et \cdot ainsi qu'entre \dots et \cdots est qu'ils ne sont pas sur la même ligne horizontale :

```
bla bla . bla bla $\cdot$

bla bla $\dots$ bla bla $\cdots$
```

bla bla . bla bla $\cdot$
bla bla $\dots$ bla bla $\cdots$

3.9 Fractions et coefficients binomiaux

Pour les fractions, on utilise la commande \frac{}{} qui prend deux arguments, dans l'ordre le numérateur et le dénominateur. Le rendu change selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle* :

```
Bla bla $\frac{1}{2}+1$ bla bla
\[
\frac{1}{2}+1
\]
```

Bla bla $\frac{1}{2}+1$ bla bla

$$\frac{1}{2}+1$$

Exercice 15

Taper les formules suivantes :

$$x^{1/3} = x^{\frac{1}{3}} = \sqrt[3]{x}$$

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \ddots}}}$$

Il peut être utile, dans certaines situations, de forcer le style d'une fraction (afficher une fraction dans le texte comme si elle était mise en évidence et vice-versa). Pour forcer le rendu *displaystyle* on peut utiliser la commande \dfrac{}{} et pour forcer le rendu *inline* on peut utiliser la commande \tfrac{}{}. Voici des exemples

```
Dans le texte: style prédéfinit $\frac{1}{2}+1$, style inline $\tfrac{1}{2}+1$, style
display $\dfrac{1}{2}+1$.

Les mêmes mais dans des formules display:
\[
\frac{1}{2}+1,\qquad \tfrac{1}{2}+1,\qquad \dfrac{1}{2}+1.
\]
```

Dans le texte : style prédéfinit $\frac{1}{2}+1$, style inline $\tfrac{1}{2}+1$, style display $\dfrac{1}{2}+1$.
Les mêmes mais dans des formules display :

$$\frac{1}{2}+1, \qquad \tfrac{1}{2}+1, \qquad \dfrac{1}{2}+1.$$

Pour les coefficients binomiaux on utilise la commande \binom{}{} (et les variantes \dbinom{}{} et \tbinom{}{}).

```
\[
\binom{n}{k}
\]
\[
\tbinom{n}{k}
\]
$\binom{n}{k}$

$\dbinom{n}{k}$
```

$$\binom{n}{k}$$

$$\tbinom{n}{k}$$

$\binom{n}{k}$

$\dbinom{n}{k}$

3.10 Lettres grecques

Pour taper les lettres grecques, il suffit de précéder le nom de la lettre par un *backslash*; par exemple α donne α. Voici une liste complète des lettres grecques disponibles sous LaTeX :

Commande	Résultat	Commande	Résultat	Commande	Résultat
\alpha	α	\xi	ξ	\Gamma	Γ
\beta	β	\pi	π	\Delta	Δ
\gamma	γ	\varpi	ϖ	\Theta	Θ
\delta	δ	\rho	ρ	\Lambda	Λ
\epsilon	ϵ	\varrho	ϱ	\Xi	Ξ
\varepsilon	ε	\sigma	σ	\Pi	Π
\zeta	ζ	\varsigma	ς	\Sigma	Σ
\eta	η	\tau	τ	\Upsilon	Υ
\theta	θ	\upsilon	υ	\Phi	Φ
\vartheta	ϑ	\chi	χ	\Psi	Ψ
\iota	ι	\phi	ϕ	\Omega	Ω
\kappa	κ	\varphi	φ		
\lambda	λ	\psi	ψ		
\mu	μ	\omega	ω		
\nu	ν				

Exercice 16

Taper la formule :

$$\frac{\pi^2}{6} + \gamma = \Gamma(n) + \sqrt[n]{1+\alpha}$$

3.11 Fonctions mathématiques

Commande	\cos	\sin	\tan	\cot	\arccos	\arcsin	\arctan	\sinh	\cosh	\tanh
Résultat	cos	sin	tan	cot	arccos	arcsin	arctan	sinh	cosh	tanh
Commande	\coth	\exp	\ln	\lg	\log	\liminf	\max	\sup	\min	\inf
Résultat	coth	exp	ln	lg	log	liminf	max	sup	min	inf
Commande	\lim	\limsup	\deg	\det	\dim	\ker	\arg	\gcd	\hom	
Résultat	lim	limsup	deg	det	dim	ker	arg	gcd	hom	

Pour mettre des bornes à ces objets, il suffit d'utiliser les commandes d'indice et/ou d'exposant. **Le placement des indices et exposants change** selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle*.

```
$\lim_{x\to 0} f(x)$
\[
\lim_{x\to 0} f(x)
\]
```

$\lim_{x\to 0} f(x)$

$$\lim_{x\to 0} f(x)$$

On peut aussi définir de nouveaux opérateurs avec

```
\DeclareMathOperator{\cotan}{cotan}
```

qui permettra d'utiliser `\cotan` pour obtenir cotan. Il y a aussi une variante étoilée pour les objets du type `\lim` ou `\max` qui prennent des bornes. Par exemple

```
\DeclareMathOperator*{\supess}{sup\,ess}
```

définira une commande `\supess` imprimant sup ess et se comportant comme `\lim` vis-à-vis des indices.

Exercice 17

Taper les formules suivantes :

$$\cos^2 + \sin^2 = 1$$

$$2^{\ln(x)} = x^{\ln(2)}$$

3.12 Grands opérateurs : intégrales, sommes, produits, etc.

Le rendu des grands opérateurs change selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle*. Certaines de ces commandes prennent, tout comme max ou lim, des bornes. Le principe est le même, on utilise des indices ou des exposants pour les taper. Le placement des indices et exposants dépend de si la formule est mise en évidence ou pas.

```
Bla bla $\int_a^b f(x) dx$ bla bla $\sum_{i=0}^n u_n$
\[
\int_a^b f(x) dx
\]
\[
\sum_{i=0}^n u_n
\]
```

Bla bla $\int_a^b f(x)\,dx$ bla bla $\sum_{i=0}^n u_n$

$$\int_a^b f(x)\,dx$$

$$\sum_{i=0}^n u_n$$

Commande	\int	\iint	\iiint	\sum	\prod
Résultat *inline*	$\int$	$\iint$	$\iiint$	$\sum$	$\prod$
Résultat *displaystyle*	$\displaystyle\int$	$\displaystyle\iint$	$\displaystyle\iiint$	$\displaystyle\sum$	$\displaystyle\prod$

Commande	\bigcup	\bigcap	\bigsqcup	\bigoplus	\bigotimes	\coprod
Résultat *inline*	$\bigcup$	$\bigcap$	$\bigsqcup$	$\bigoplus$	$\bigotimes$	$\coprod$
Résultat *displaystyle*	$\displaystyle\bigcup$	$\displaystyle\bigcap$	$\displaystyle\bigsqcup$	$\displaystyle\bigoplus$	$\displaystyle\bigotimes$	$\displaystyle\coprod$

Pour mettre plusieurs lignes dans les indices, il faut utiliser \substack{} ; à l'intérieur de l'argument de \substack, on passe à la ligne avec \\.

```
\[
L_i(x)=\prod_{\substack{j=0\\j\neq i}}^n\frac{x-x_j}{x_i-x_j}
\]
```

$$L_i(x) = \prod_{\substack{j=0\\j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}$$

Exercice 18

Taper les formules suivantes :

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

$$\int_0^1 -\frac{\ln(1-t)}{t}\,dt \approx 1{,}64493$$

$$\max_{\substack{x,y\in E\\ x\cdot y=0}} \varphi(x)$$

3.13 Accents mathématiques

Pour mettre un accent sur une lettre seule on pourra utiliser l'un de ces commandes :

Commande	\tilde{a}	\vec{a}	\hat{a}	\check{a}	\mathring{a}	\dot{a}	\ddot{a}
Résultat	$\tilde{a}$	$\vec{a}$	$\hat{a}$	$\check{a}$	$\mathring{a}$	$\dot{a}$	$\ddot{a}$
Commande	\dddot{a}	\ddddot{a}	\acute{a}	\breve{a}	\grave{a}	\bar{a}	
Résultat	$\dddot{a}$	$\ddddot{a}$	$\acute{a}$	$\breve{a}$	$\grave{a}$	$\bar{a}$	

Il existe cependant des accents extensibles pour couvrir plusieurs lettres :

\widetilde{abc} $\widetilde{abc}$	\widehat{abc} $\widehat{abc}$	\overline{abc} $\overline{abc}$	\underline{abc} $\underline{abc}$
\overbrace{abc} $\overbrace{abc}$	\underbrace{abc} $\underbrace{abc}$	\overrightarrow{abc} $\overrightarrow{abc}$	\underrightarrow{abc} $\underrightarrow{abc}$
\overleftarrow{abc} $\overleftarrow{abc}$	\underleftarrow{abc} $\underleftarrow{abc}$	\overleftrightarrow{abc} $\overleftrightarrow{abc}$	\underleftrightarrow{abc} $\underleftrightarrow{abc}$

Pour des vecteurs (extensibles), mieux utiliser la commande \vv{} du package esvect :

```
$\vec{u}$, $\vv{u}$, $\vec{AB}$, $\vv{AB}$
```

$\vec{u}$, $\vv{u}$, $\vec{AB}$, $\vv{AB}$

Pour \underbrace il est possible de placer du matériel en-dessous en utilisant _ et pour \overbrace du matériel au-dessus en utilisant ^.

```
\[
\underbrace{x^3 + x^2 + x + 1}_{\to 0}
\qquad
\overbrace{x^3 + x^2 + x + 1}^{\to 0}
\]
```

$$\underbrace{x^3+x^2+x+1}_{\to 0} \qquad \overbrace{x^3+x^2+x+1}^{\to 0}$$

Exercice 19

Écrire la formule suivante :

$$\overrightarrow{OM} = \underbrace{O+\vec{u}}_{\text{point+vecteur}}$$

3.14 Délimiteurs

Commande	Résultat	Commande	Résultat	Commande	Résultat
(	(	\lvert	|	\lceil	⌈
)	)	\rvert	|	\rceil	⌉
[	[	\lVert	‖	\langle	⟨
]	]	\rVert	‖	\rangle	⟩
\{	{	\lfloor	⌊	/	/
\}	}	\rfloot	⌋	\backslash	\

.

Il est important de comprendre que, même si `\lvert` et `\rvert` se ressemblent, ils ne peuvent pas être interchangés : `\lvert` doit toujours être utilisé pour ouvrir et `\rvert` pour refermer. Par exemple, $|x|$ se tape `$\lvert x \rvert$`. Les seuls délimiteurs à n'être ni ouvrant ni fermant mais médian sont le slash / et l'anti-slash \.

Exercice 20

Taper les formules suivantes :

$$\|x\| = 1 \iff \langle x, x\rangle = 1$$

$$|\{1,2,\dots,n\}| = n$$

$$\lfloor x^2+\epsilon \rfloor = \lceil \sqrt{y}+\delta \rceil$$

Pour avoir des délimiteurs qui sont de la même taille que ce qu'ils entourent, il faut précéder le délimiteur ouvrant par `\left` et le délimiteur fermant par `\right`.

```
\[
\left(1+\frac{1}{n}\right)^n
\]
```

$$\left(1+\frac{1}{n}\right)^n$$

S'il n'y a qu'un délimiteur, il faut mettre un délimiteur "vide" indiqué par un point :

```
$PV=nRT$ donc
\[
\left.\frac{\partial P}{\partial T}\right|_{V}=\frac{nR}{V}
\]
```

$PV = nRT$ donc

$$\left.\frac{\partial P}{\partial T}\right|_V = \frac{nR}{V}$$

Exercice 21

Taper les formules suivantes :

$$\left[\sum_{n=1}^{N} u_n\right]^2 = N^2 + N + 1$$

$$\left[1 + \left(\int_0^{\sqrt{2}} f\right)^2\right] = \gamma$$

big Big BBig etc, voir «courte intro»

On peut aussi utiliser `\middle` pour mettre un délimiteur médian au milieu d'un couple `\left` et `\right`, comme dans la définition d'un ensemble :

$$\left\{ x \middle| x^2 < \frac{1}{2} \right\}$$

```
\[
\left\{ x \middle| x^2<\frac{1}{2} \right\}
\]
```

On note qu'ici la barre verticale est trop proche de ce qu'il y a autour d'elle et il faut rajouter à la main deux petits espaces avec la commande `\,` :

$$\left\{ x \,\middle|\, x^2 < \frac{1}{2} \right\}$$

```
\[
\left\{ x \,\middle|\, x^2<\frac{1}{2} \right\}
\]
```

Pour écrire plus simplement les ensembles, on peut utiliser le package `braket` : dans le préambule on ajoute

```
\usepackage{braket}
```

et dans le corps du document on utilise la commande `\Set{}`. L'exemple précédent s'écrit alors

$$\left\{ x \,\middle|\, x^2 < \frac{1}{2} \right\}$$

```
\[
\Set{ x | x^2<\frac{1}{2} }
\]
```

3.15 Alphabets mathématiques

Voici un résumé des alphabets mathématiques disponibles après importation dans le préambule des packages `amssymb` et `amsfonts` (on remplacera la lettre C par une lettre majuscule quelconque et la lettre x par une lettre minuscule quelconque) :

Commande	Résultat
`\mathbb{C}`	$\mathbb{C}$
`\mathcal{C}`	$\mathcal{C}$
`\mathfrak{C}`	$\mathfrak{C}$

Commande	Résultat
`\mathrm{x}`	x
`\mathbf{x}`	$\mathbf{x}$
`\mathtt{x}`	$\mathtt{x}$
`\mathsf{x}`	x
`\mathit{PGL}`	PGL

À noter que le résultat obtenu ici avec la commande `\mathcal{C}` n'est pas le même que dans nos salles de TP (qui est obtenu avec la police par défaut Computer Modern du package `lmodern`). En effet, pour cet ouvrage j'utilise l'extension `fourier` qui modifie le résultat. Si l'on utilise la police par défaut et que l'on souhaite faire des majuscules manuscrites, on peut utiliser le package `!`mathrsfs`!` et la commande `\mathscr{C}`.

Ne pas confondre `\mathrm` et `\text` : le premier est pour mettre des maths en romain tandis que le second est pour insérer du texte dans une formule.

Attention : pour mettre en gras une lettre grecque, on ne peut pas utiliser `\mathbf{}` mais il faut utiliser `\boldsymbol{}` :

```
\[
\sigma \qquad \mathbf{\sigma} \qquad \boldsymbol{\sigma}
\]
```

$$\sigma \qquad \sigma \qquad \boldsymbol{\sigma}$$

Exercice 22

Reproduire le texte suivant :

Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction d'ensemble de définition $\mathscr{D}_f$ et de courbe représentative $\mathscr{C}_f$. Soit $\mathscr{F}$ une famille libre de vecteurs.

Exercice 23

Taper les formules suivantes :

$$\{a + ib \in \mathbb{C} \mid a < b\}$$

$$\mathscr{L} f = \int_a^b f \, \mathrm{d}t$$

3.16 Tableaux mathématiques

Pour composer des tableaux mathématiques on utilise l'environnement `{array}` qui fonctionne de manière similaire à l'environnement `{tabular}` :

$f(x)$	$f'(x)$
x^n	nx^{n-1}
$\ln(x)$	$\frac{1}{x}$

```
\[
\begin{array}{|c|c|}
\hline
f(x)    &   f'(x)\\
\hline
x^n     &   nx^{n-1}\\
\ln(x)  &   \frac{1}{x}\\
\hline
\end{array}
\]
```

Exercice 24

Reproduire la formule suivante :

$$\left.\begin{array}{l} a \in \mathbb{C} \\ a \notin \mathbb{R} \end{array}\right] \Longrightarrow a \in \mathbb{C} \setminus \mathbb{R}$$

3.17 Distinction de cas

Pour définir une fonction par morceaux on peut utiliser l'environnement `{array}` mais le package `amsmath` met à disposition un environnement dédié, l'environnement `{cases}`, qui s'utilise de la façon suivante

$$H(x) = \begin{cases} 0 & \text{si } x < 0, \\ 1 & \text{si } x \geq 0. \end{cases}$$

```
\[
H(x) =
\begin{cases}
0 & \text{si $x < 0$,} \\
1 & \text{si $x \geq 0$.}
\end{cases}
\]
```

3.18 Matrices

On tape les matrices comme des tableaux, sauf qu'on n'utilise plus l'environnement `{array}`, mais les environnements `{matrix}`, `{pmatrix}`, etc.

$$\begin{matrix} a & b \\ c & d \end{matrix}$$
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
$$\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$$
$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$
$$\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

```
\[
\begin{matrix} a & b \\ c & d \end{matrix}
\]
\[
\begin{pmatrix} a & b \\ c & d \end{pmatrix}
\]
\[
\begin{bmatrix} a & b \\ c & d \end{bmatrix}
\]
\[
\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}
\]
\[
\begin{vmatrix} a & b \\ c & d \end{vmatrix}
\]
\[
\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}
\]
```

Les matrices peuvent avoir jusqu'à 10 colonnes ; si jamais il y a besoin d'en avoir plus (par exemple, 15 colonnes), rajouter dans le préambule, après avoir chargé tous les packages, la ligne

```
\setcounter{MaxMatrixCols}{15}
```

Exercice 25

Reproduire la matrice suivante :

$$\mathbb{M} = \begin{pmatrix} m_{1,1} & \dots & m_{1,n} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \dots & m_{n,n} \end{pmatrix}$$

3.19 Formules sur plusieurs lignes et alignements

Dans une formule *displaystyle* l'«aller à la ligne» est interdit (car le changement de paragraphe est interdit en mode mathématique). On dispose alors d'environnements spécifiques pour écrire une formule sur plusieurs lignes : les environnements `{align}`, `{gather}` et `{multline}` du package `amsmath`.

- L'environnement `{align}` (ou sa variante non numérotée `{align*}`) permet d'aligner plusieurs signes d'égalité à l'intérieur d'une même formule :

```
\begin{align}
A & = B \\
  & = C+D \\
  & = E
\end{align}
```

$$A = B \tag{3.3}$$
$$= C + D \tag{3.4}$$
$$= E \tag{3.5}$$

On peut au besoin mettre plusieurs équations en colonne :

```
\begin{align*}
A & = B   & E & = F & I & = J \\
  & = C+C &   & = G &   & = K \\
  & = D   &   &     &   &
\end{align*}
```

$$A = B \qquad E = F \qquad I = J$$
$$= C + C \qquad = G \qquad = K$$
$$= D$$

Ne jamais mettre de \\ sur la dernière ligne d'un {align}.

- L'environnement {gather} (ou sa variante non numérotée {gather*}) est fait pour écrire plusieurs équations les unes en-dessous des autres et centrée sur chaque ligne :

```
\begin{gather}
A   = B+C \\
D+E = F \\
G+H = I+J
\end{gather}
```

$$A = B + C \tag{3.6}$$
$$D + E = F \tag{3.7}$$
$$G + H = I + J \tag{3.8}$$

Ne jamais mettre de \\ sur la dernière ligne d'un {gather}.

- L'environnement {multline} (ou sa variante non numérotée {multline*}) permet de découper une formule trop grande en plusieurs morceaux sans contrôle précis de l'alignement. Voici un exemple :

```
\begin{multline}
A+B+C+D+E+F+G+H+I+J\\
+A+B+C+D+E+F+G+H+I+J\\
+A+B+C+D+E+F+G+H+I+J\\
+A+B+C+D+E+F+G+H+I+J
\end{multline}
```

$$A+B+C+D+E+F+G+H+I+J$$
$$+A+B+C+D+E+F+G+H+I+J$$
$$+A+B+C+D+E+F+G+H+I+J$$
$$+A+B+C+D+E+F+G+H+I+J \tag{3.9}$$

Ne jamais mettre de \\ sur la dernière ligne d'un {gather}.

Finalement, on peut vouloir numéroter uniquement certaines lignes d'une équation. Il y a alors la commande \notag qui permet de désactiver la numérotation sur une ligne. Voici un exemple :

```
\begin{align}
A & = B \\
& = C \notag \\
& = D \notag
\end{align}
```

$$A = B \tag{3.10}$$
$$= C$$
$$= D$$

Le package amsmath défini d'autres environnements, vous pouvez les utiliser (après avoir lu la documentation du package et compris dans quels cas il faut utiliser un environnement plutôt qu'un autre).[4]

[4] Attention : l'utilisation de l'environnement qnarrayeqnarray pour mettre en forme une formule sur plusieurs lignes, qu'on peut trouver dans certains vieux manuels, sera considéré comme une faute grave et sévèrement sanctionné. Cet environnement a plusieurs défauts bien illustré dans ce document : Avoid eqnarray! disponible à l'adresse http://tug.org/pracjourn/2006-4/madsen/.

Exercice 26

Reproduire les formules suivantes :

$$f : \mathbb{R} \to \mathbb{R} \qquad g : \mathbb{R} \to \mathbb{R}$$
$$x \mapsto x^2 \qquad x \mapsto \sqrt{x}$$

et

$$\begin{aligned}\left|\int_a^b (f+g)\right| &= \left|\int_a^b f + \int_a^b g\right| \\ &\leq \left|\int_a^b f\right| + \left|\int_a^b g\right| \\ &\leq \int_a^b |f| + \int_a^b |g|\end{aligned}$$

3.20 Flèches extensibles

Le package `amsmath` dispose de deux flèches extensibles, `\xleftarrow[]{}` et `\xrightarrow[]{}`. Pour mettre une formule dessus, on utilise l'argument obligatoire tandis que pour mettre une formule dessous, on utilise l'argument optionnel :

$$f(x) \xrightarrow{\text{d'après } (H)} a$$
$$f(x) \xrightarrow[x\to 0]{} a$$
$$f(x) \xrightarrow[x\to 0]{\text{d'après } (H)} a$$

```
\[
f(x) \xrightarrow{\text{d'après }(H)} a
\]
\[
f(x) \xrightarrow[x\to 0]{} a
\]
\[
f(x) \xrightarrow[x\to 0]{\text{d'après }(H)} a
\]
```

3.21 Modules de congruences

Selon l'apparence voulue, il y a trois façon d'écrire les modules de congruence :

$a \equiv b \mod m$
$a \equiv b \pmod m$
$a \equiv b \pod m$

```
$a \equiv b \mod m$

$a \equiv b \pmod m$

$a \equiv b \pod m$
```

Il y a aussi la commande `\bmod` qu'on peut utiliser dans le contexte suivant

$\gcd(n, m \bmod n)$

```
$\gcd(n,m\bmod n)$
```

Exercice 27

Reproduire la formule suivante :

$$\sum_{n \equiv a \bmod p} \frac{1}{n} < +\infty.$$

3.22 Placer au-dessus ou en-dessous

Les commandes `\underset{en-dessous}{symbole}` et `\overset{au-dessus}{symbole}` permettent de placer du matériel arbitraire en-dessous ou au-dessus de n'importe quel symbole.

$$\overset{\text{déf}}{=}$$

$$x \underset{x\to 0}{\in} o(x^2)$$

```
\[
\overset{\text{déf}}{=}
\]
\[
x \underset{x \to 0}{\in} o(x^2)
\]
```

3.23 Exercices de synthèse

Exercice 28

Reproduire les formules suivantes :

$$\sqrt{\sum_{n=0}^{+\infty} u_n} = \left(\int_a^b f\,\mathrm{d}t\right)^2 + \gamma \times \frac{\pi^2}{6}$$

$$\lim_{x\to 0^+} f(x) = \sin\left|\frac{1-\sqrt[2]{3}}{2}\right| \quad \text{mais} \quad f(0) = \ln^2(3)$$

$$\lim_{\substack{x\to 0\\ x>0}} \frac{\left\lfloor (\sin(x))^{\cos(x)} - (\cos x)^{\sin(x)} \right\rfloor}{x^3}$$

$$\forall x \in \mathbb{R}, \quad f(x) = 0 \iff x^{1/3} + \ln(\tan(x)) = \omega$$

Exercice 29

En utilisant des commandes personnelles judicieusement choisies, taper les formules suivantes :

$$\left|\left\{ x \in \mathbb{Z} \;\middle|\; x^2 < 2 \right\}\right| = \bigcup_{n\in\mathbb{N}^*} \left|\left\{ x \in \mathbb{Z} \;\middle|\; x^2 < 2 - \frac{1}{n} \right\}\right|$$

$$\mathbb{S}^1 = \left\{ x \in \mathbb{R}^2 \;\middle|\; \|x\| = 1 \right\}$$

$$\sum_{n=1}^{+\infty} \frac{1}{n^2 C_{2n}^n} = \frac{\zeta(2)}{3} = \frac{\pi^2}{18}$$

$$S(x) = \int_0^x \frac{\sin(x)}{x}\,\mathrm{d}x \quad \text{avec} \quad \lim_{x\to+\infty} S(x) = \frac{\pi}{2}$$

3.24 Pour aller plus loin

- Pour plus d'information sur la rédaction des mathématiques avec LaTeX on pourra consulter le fichier «Mathmode» écrit par H. Voss disponible en `pdf` à l'adresse
  https://texdoc.org/serve/mathtools/0
- Pour tous les symboles disponibles en LaTeX , voir «*The Comprehensive LATEX Symbol List*» disponible en `pdf` à l'adresse
  http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf

- Vous pouvez dessiner le symbole et detexify vous dira quel package charger et la commande pour obtenir ce symbole :
  http://detexify.kirelabs.org/classify.html

Faire attention au(x) package(s) nécessaire(s) à un symbole, tous les packages ne sont pas forcément compatibles avec notre préambule ni forcément installés.

☞ Récapitulatif sur les caractères ambigus

Il y a un certain nombre de caractères ambigus dans un pdf qui sont générés par des commande LaTeX différentes. Par exemple, dans les deux formules «$d \mid n$» et «$|x|$», la barre verticale a trois significations distinctes donc doit être tapée différemment à chaque fois pour obtenir un espacement correct.

- Le caractère |

Signification	Exemple	Code LaTeX
Valeur absolue	$\lvert -x \rvert$	`$\lvert -x \rvert$`
Divise	$d \mid n$	`$d \mid n$`
Tel que	$x \mid \sin(x) = 0$	`$x \mid \sin(x)=0$`
Tel que (dans un ensemble)	$\{x \mid \sin(x) = 0\}$	`$\Set{x \mid \sin(x)=0}$`
Restreint à	$f\vert_{\mathbb{N}}$	`$f\vert_\N$`

- Le caractère ‖

Signification	Exemple	Code LaTeX
Norme	$\lVert -x \rVert$	`$\lVert -x \rVert$`
Divise exactement	$p^2 \parallel n$	`$p^2 \parallel n$`
Parallèle à	$(AB) \parallel (CD)$	`$(AB)\parallel(CD)$`

- Le caractère :

Signification	Exemple	Code LaTeX
Deux points	$f\colon X \to Y$	`$f\colon X\to Y$`
	$\mathscr{C}\colon x^2+y^2=1$	`$\mathcal{C} \colon x^2+y^2=1$`
Indice… dans	$[G:H]$	`$[G:H]$`
Tel que	$\{x : \sin(x)=0\}$	`$\Set{x : \sin(x)=0}$`
Point projectif	$[x_1 : \ldots : x_n]$	`$[x_1 : \dots : x_n]$`

4 Faire des bibliographies avec LaTeX

Pour produire une bibliographie avec LaTeX , il est bien-sûr possible de confectionner une bibliographie à la main, mais il est beaucoup plus efficace de recourir au programme BibTeX. BibTeX trie automatique des entrées, présentation homogène de toutes les entrées, changement instantané de style de bibliographie, gestion de la casse, réutilisation des entrées dans d'autres documents, choix automatique de l'étiquette la plus longue, etc.

4.1 Structure d'un fichier `.bib`

Les fichiers `.bib` sont des listes de documents dont certains (éventuellement pas tous à chaque fois) sont à citer. Le type de ces documents (article, livre, documentation technique, etc.) et leurs caractéristiques (titre, auteur(s), année, etc.) sont saisis selon un format à respecter scrupuleusement. Chacun de ces documents est identifié au moyen d'un identifiant, appelé *clé*. Voici un exemple de fichier `.bib` constituant une base bibliographique :

```
@Article{El03,
 author = {Loughran, Ellen},
 title = {Tentative {B}eginnings: {M}ontaigne {R}ewrites {H}is {E}arly {E}ssays},
 journaltitle = {Neophilologus},
 date = {2003},
 volume = {87},
 number = {3},
 pages = {371383}
}

@Book{lcfr,
 author = {Mittelbach, Frank and Goossens, Michel},
 title = {\LaTeX{} {C}ompanion},
 publisher = {Pearson Education France},
 date = {200602},
 location = {Paris},
 pagetotal = {1116},
 edition = {2}
}

@Manual{Leh,
 title = {The \texttt{biblatex} package},
 subtitle = {Programmable bibliographies and citations},
 author = {Lehman, Philipp},
 version = {1.7},
 date = {20111113},
 url = {http://tug.ctan.org/pkg/biblatex}
}
```

Pour créer sa base de donnée on pourra utiliser `google scholar` comme suit :

— chercher sur `https://scholar.google.fr` les livres/articles que vous voulez insérer dans votre bibliographie

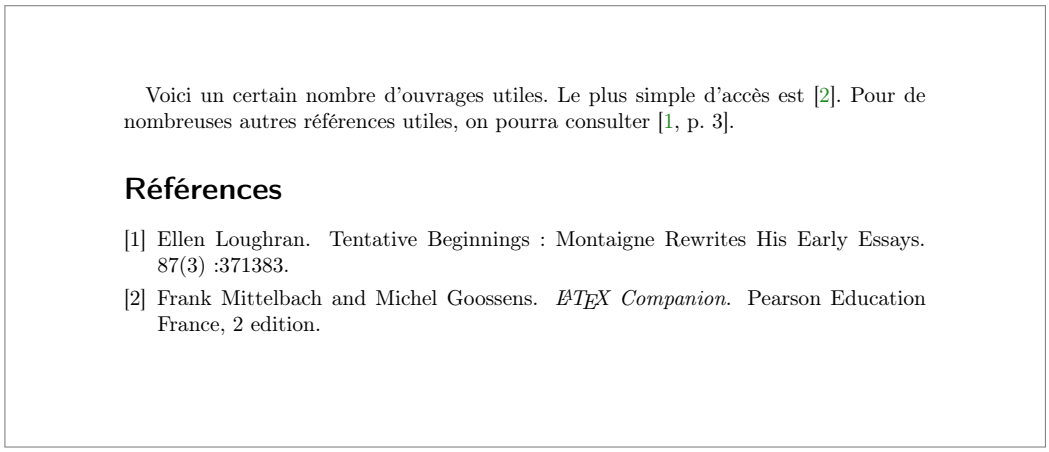


FIGURE 4.1 – Exemple de document avec bibliographie.

— cliquer sur les deux chevrons ≫ après la description du livre choisi
— cliquer sur les guillemets
— choisir le format `bibtex`
— copier le tout dans un fichier `biblio.bib`

4.2 Fichier .tex

Analysons le document à la figure 4.1.

1. Dans les deux premières lignes, il y a des références aux éléments de la bibliographie. Ces références se font grâce à la commande `\cite`. Chaque élément de la bibliographie est identifié par une clef, disons `lcfr` et c'est cette clef que l'on passe en argument à `\cite` :

   ```
   \cite{lcfr}
   ```

   Cela imprimera automatiquement l'étiquette utilisée dans la bibliographie pour cette référence (typiquement, ce sera un numéro, disons [2], mais ce pourrait aussi être les initiales, disons [MG], selon le style bibliographique choisi) et le lien sera cliquable. Si jamais on veut spécifier un numéro de page ou un théorème particulier, `\cite` peut prendre un argument optionnel :

   ```
   \cite[p.~3]{El03}
   ```

   donnera [1, p. 3] si l'étiquette est [1] et [Lou, p. 3] si c'est [Lou].

2. À l'endroit où l'on veut faire apparaître la bibliographie on écrira les commandes `\bibliographystyle` et `\bibliography` (la première spécifie le stylede la bibliographie et la seconde le fichier de bibliographie). Par exemple, les lignes

   ```
   \bibliographystyle{plain}
   \bibliography{biblio}
   ```

   feront que le style `plain` sera utilisé et que c'est le fichier `biblio.bib` qui sera utilisé pour générer la bibliographie.

   À la figure 4.2 on voit un exemple d'un même document avec le style `plain` et avec le style `alpha`. [1] Si on écrit un document en français, on veillera à utiliser un style adapté à la langue française (comme par exemple le style `plain-fr`).

[1] On pourra utiliser par exemple `plain`, `alpha`, `abbrv`, `acm`, `ieetr`, `siam`.

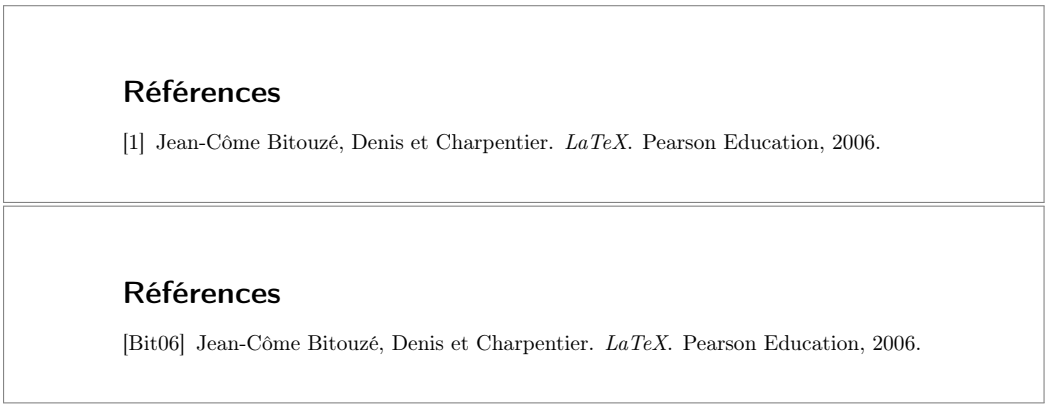


FIGURE 4.2 – Exemple d'un même document avec le style `plain` (en haut) puis avec le style `alpha` (en bas).

Exercice 30

Reproduire le document donné à la figure 4.1.

4.3 Compilation

Il faut faire un certain nombre de compilations :

— **une** première **compilation** avec `pdflatex` (touche F6 avec `Texmaker`) ; cette compilation écrit les informations utiles à la construction de la bibliographie dans le fichier `.aux` ;

— ensuite **on compile** avec `bibtex` (touche F11 sous `Texmaker`) ; cette compilation produit un fichier `.bbl` qui contient un environnement `{thebibliography}` et qui sera inclue dans le document à la prochaine compilation ;

— enfin **on compile deux fois** de suite avec `pdflatex` (touche F6 deux fois avec `Texmaker` puis touche F7 pour visualiser le fichier `pdf` produit).

Attention : seul les documents cités seront affichés dans la bibliographie. [2]

Exercice 31

Écrire un document qui contient au moins trois des livres/articles de votre base de données bibliographiques.

4.4 Pour aller plus loin

- Pour gérer la base de donnée `bibtex` on pourra utiliser le logiciel multi-plateforme `Jabref`.
- Pour une mise en page personnalisée des citations et de la bibliographie on pourra utiliser le package `biblatex` :
  - le package officiel
    `https://ctan.org/pkg/biblatex`
  - "BibLaTeX expliqué à Mademoiselle Michu, étudiante en Sciences Humaines"
    `http://bertrandmasson.free.fr/index.php?article27/`

[2] Il est possible, mais déconseillé, d'importer tous les documents du fichier `bib` en écrivant avant l'importation de la bibliographie la commande `\nocite{*}`

5 Présentations vidéo-projetées

`Beamer` est une classe comme `scrartcl`, `scrreprt`... dont le but est de réaliser des présentations, c'est à dire une série de diapositives destinées à être projetées. `Beamer` présente plusieurs avantages sur d'autre logiciel comme `Impress` de la suite `OpenOffice` (`LibreOffice`) ou `PowerPoint` de `Microsoft`. Tout d'abord c'est du LaTeX donc une qualité typographique irréprochable. De plus, tous ce qu'on a appris sur LaTeX est utilisable dans les présentations. Le document de sortie est un `pdf` donc sera lisible sans déformation sur n'importe quel type de machine et avec n'importe quel OS (Linux, Windows, Mac, Android).

5.1 Introduction

Comme toutes les autres classes, `beamer` se charge par

```
\documentclass[options]{beamer}
```

Il faut savoir que `beamer` charge automatiquement les packages `hyperref`, `xcolor` et `enumerate`. Si on veut leurs adjoindre des options on doit procéder de la façon suivante :

```
\PassOptionsToPackage{table,dvipsnames,svgnames}{xcolor}
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}
```

L'option `pdfpagemode=FullScreen` d'`hyperef` permet de lancer automatiquement en mode plein écran la présentation.

Normalement, on passe les options à `xcolor` de la même manière qu'à `hyperref` :

```
\documentclass
[
hyperref={pdfpagemode=FullScreen,colorlinks=false},
xcolor={table,dvipsnames,svgnames}
]{beamer}
```

Cependant, il y a un bug dans la version 3.65 de beamer et l'instruction `\PassOptionsToPackage` contourne ce bug. Les anciennes versions ainsi que les suivantes aussi ont été corrigées.

5.1.1 Ratio

Les dimensions de la diapositive dans `beamer` sont de 128 mm sur 96 mm (ratio de 4 : 3). Il est cependant préférable de modifier ce comportement pour avoir un ratio plus adapté aux écran d'aujourd'hui, à savoir 16 : 9.

Pour cela, on utilise l'option de classe `aspectratio` :

```
\documentclass[aspectratio=169]{beamer}
```

D'autres valeurs possibles sont 1610 pour 16 : 10, 149 pour 14 : 9, 54 pour 5 : 4, 43 pour 4 : 3 et 32 pour 3 : 2.

5.2 Choix du thème et création de la première diapositive

Le but de ces notes est de réaliser le plus simplement possible une présentation et pas de montrer toutes les possibilités de `beamer`, on va donc charger un modèle. On trouvera une galerie des thèmes disponibles [1] sur ce site

[1] Par exemple parmi les thèmes complets on a :

http://mcclinews.free.fr/latex/beamergalerie.php

On crée chaque diapositive avec l'environnement {frame}. Voici un exemple à la figure 5.1 (avec le thème Madrid) obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}
%\documentclass[aspectratio=169,hyperref={pdfpagemode=FullScreen,colorlinks=false}]{
beamer}

 \usepackage{concrete} % une police qui va bien pour l'écran
 \usepackage[latin1]{inputenc} % si windows
%\usepackage[utf8]{inputenc}   % si linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Madrid}

 \title{Le titre de la présentation}
 \author{G. \textsc{Faccanoni}}
 \institute{IMATH-UTLN}

 \begin{document}

 \begin{frame}[plain]
 \maketitle
 \end{frame}

 \begin{frame}
 Le texte de ma diapo.
 \end{frame}

\end{document}
```

.

À la figure 5.2 les mêmes diapositives avec un aspectratio 16 : 9.
À la figure 5.3 les mêmes diapositives avec un aspectratio 16 : 9 et le thème CambridgeUS.

— Pittsburgh, Rochester, Bergen, Boadilla, Madrid, AnnArbor, CambridgeUS,
— Montpellier, Antibes, JuanLesPins,
— Goettingen, Hannover, Marburg, Berkeley, Paloalto
— Singapore, Szeged, Berlin, Ilmenau, Dresden, Darmstadt, Frankfurt
— Malmoe, Copenhagen, Luebeck, Warsaw

(a) Diapo initiale (couverture).

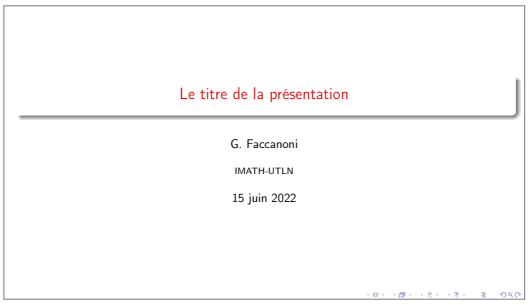
(b) Diapo avec titre.

FIGURE 5.1 – Exemple de diapositives `beamer` (thème `Madrid`).

(a) Diapo initiale (couverture).

(b) Diapo avec titre.

FIGURE 5.2 – Exemple de diapositives `beamer` (thème `Madrid`) avec un `aspectratio` 16 : 9.



(a) Diapo initiale (couverture).

(b) Diapo avec titre.

FIGURE 5.3 – Exemple de diapositives `beamer` (thème `CambridgeUS`) avec un `aspectratio` 16 : 9.

On peut utiliser le même sectionnement que sous LaTeX pour regrouper plusieurs diapositives par section ou sous-section. Voici un exemple à la figure 5.4 obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Antibes}

\begin{document}

 \begin{frame}
 \frametitle{Outline}
 \tableofcontents
 \end{frame}

\section{Titre de la section}

 \subsection{Titre de la sous-section 1}

 \begin{frame}
 \frametitle{Titre à la diapo}
 Première diapo de le sous section 1
 \end{frame}

 \begin{frame}
 \frametitle{Titre à la diapo}
 Deuxième diapo de le sous section 1
 \end{frame}

 \subsection{Titre de la sous-section 2}

 \begin{frame}
 \frametitle{Titre à la diapo}
 Première diapo de le sous section 1
 \end{frame}

\end{document}
```

Il ne faut pas confondre section et titre de la diapositive. Le sectionnement permet de regrouper plusieurs diapositives sous un même entête. Si le titre de la diapositive apparaît sur la diapositive elle-même, le sectionnement pour être visible doit être utilisé avec un thème affichant le sommaire (comme `Hannover` ou `Antibes`) ou afficher une diapositive avec la table de matière avant chaque début de section.

Attention : un environnement `frame` ne doit contenir ni sections ni sous-sections.

5.3 Les blocs

En plus de tous les objets LaTeX (images, tableau, listes...) on peut insérer dans les diapositives des objets propres à `beamer` comme par exemple les `block`. Leur aspect dépend du thème choisi. Voici un exemple à la figure 5.5a avec le thème `Warsaw` obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}
```

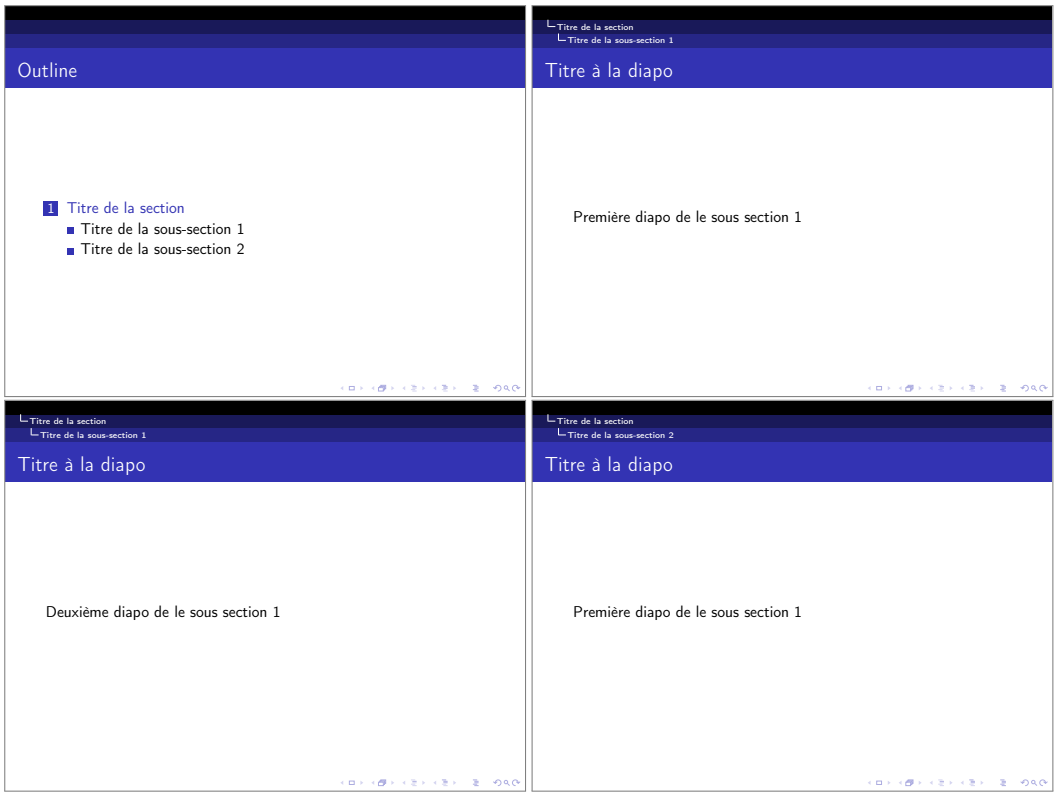


FIGURE 5.4 – Exemple de diapositives regroupées par sections

```
 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

 \begin{frame}
 \frametitle{Titre à la diapo}
 Bla bla
 \begin{block}{Titre du block}
 Le texte du block.
 \end{block}
 Bla bla
 \begin{block}{}
 Le texte du block.
 \end{block}
 Bla bla
 \end{frame}

\end{document}
```

Il existe deux autres type de blocs, dont la seule différence réside dans la couleur : le bloc `alert` et le bloc `example`. Voici un exemple à la figure 5.5b avec le thème `Warsaw` obtenu avec le code

(a) Blocs avec ou sans titre.

(b) Blocs de type `alert` et `example`.

FIGURE 5.5 – Exemple de diapositives avec blocs (thème `Warsaw`).

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

 \begin{frame}
 \frametitle{Titre de la diapo}

 Bla bla

 \begin{alertblock}{Titre du block}
 Le texte du block.
 \end{alertblock}

 Bla bla

 \begin{alertblock}{}
 Le texte du block.
 \end{alertblock}

 Bla bla

 \begin{exampleblock}{Titre du block}
 Le texte du block.
 \end{exampleblock}

 Bla bla

 \begin{exampleblock}{}
 Le texte du block.
 \end{exampleblock}
```

```
 \end{frame}

\end{document}
```

Il existe également plusieurs type de blocs prédéfinit : les définitions, les exemples, les démonstrations, les théorèmes. La traduction des titres de ces blocs n'est pas prise en compte par `babel` mais par le package `translator` qui est automatiquement chargé par `beamer` mais auquel il faut passer l'option `French` en mettant dans le préambule le deux instructions `\uselanguage{French}\languagepath{French}`. Voici un exemple à la figure 5.6 avec le thème `Warsaw` obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

 \uselanguage{French}
 \languagepath{French}

\begin{document}

 \begin{frame}
 \frametitle{Titre de la diapo}

 Bla bla

 \begin{definition}
 Le texte de la définition.
 \end{definition}

 Bla bla

 \begin{example}
 Le texte de l'exemple.
 \end{example}

 Bla bla

 \begin{theorem}
 Le texte du théorème.
 \end{theorem}

 \begin{proof}
 Le texte de la démonstration.
 \end{proof}

 \end{frame}

\end{document}
```

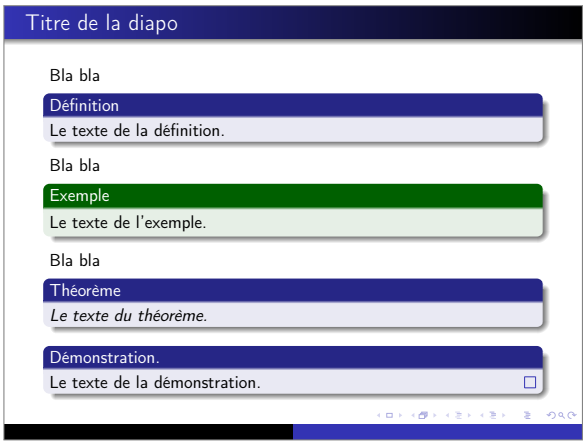
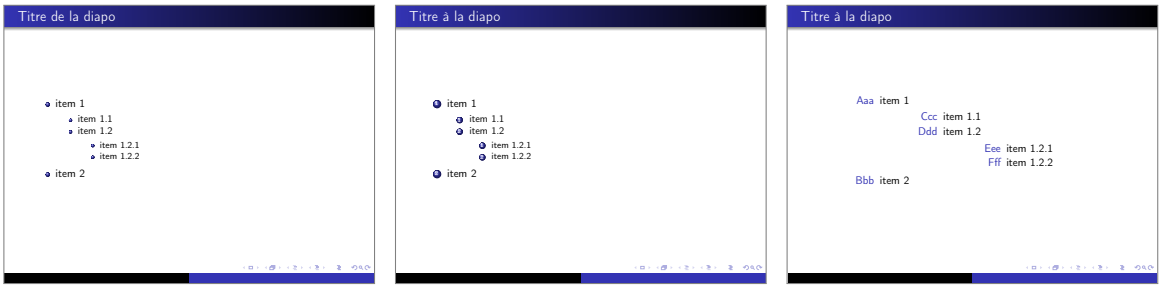


FIGURE 5.6 – Exemple de diapositives avec blocs prédéfinis (thème `Warsaw`).

5.4 Les listes

L'aspect des listes est modifié par `beamer` et dépend du thème employé. `beamer` chargeant automatiquement le package `enumerate`, on peut employer directement les options de ce dernier pour modifier les énumérations. Voici à la figure 5.7 plusieurs listes utilisant respectivement les environnements `itemize`, `enumerate` et `description` avec les valeurs par défaut et le thème `Warsaw` obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

 \begin{itemize}
 \item item 1
    \begin{itemize}
    \item item 1.1
    \item item 1.2
        \begin{itemize}
        \item item 1.2.1
        \item item 1.2.2
        \end{itemize}
    \end{itemize}
 \item item 2
 \end{itemize}

\end{frame}


\begin{frame}
```



(a) Diapo avec `itemize`. (b) Diapo avec `enumerate`. (c) Diapo avec `description`.

FIGURE 5.7 – Exemple de diapositive avec listes (thème `Warsaw`).

```
\frametitle{Titre à la diapo}

 \begin{enumerate}
 \item item 1
    \begin{enumerate}
    \item item 1.1
    \item item 1.2
        \begin{enumerate}
        \item item 1.2.1
        \item item 1.2.2
        \end{enumerate}
    \end{enumerate}
 \item item 2
 \end{enumerate}

\end{frame}


\begin{frame}
\frametitle{Titre à la diapo}

 \begin{description}
 \item[Aaa] item 1
    \begin{description}
    \item[Ccc] item 1.1
    \item[Ddd] item 1.2
        \begin{description}
        \item[Eee] item 1.2.1
        \item[Fff] item 1.2.2
        \end{description}
    \end{description}
 \item[Bbb] item 2
 \end{description}

\end{frame}

\end{document}
```

5.5 Ajouter des colonnes

On peut créer des colonnes de cette manière (attention au «s» de l'environnement externe `columns`) :

```
\begin{columns}
 \begin{column}{dimension de la première colonne}
 Le texte de la colonne
 \end{column}
 \begin{column}{dimension de la deuxièeme colonne}
 Le texte de la colonne
 \end{column}
\end{columns}
```

On utilise autant de `\begin{column}{dimension}...\end{column}` que l'on souhaite de colonnes. Pour les dimensions, mieux vaut utiliser des valeurs relatives comme `{0.5\textwidth}`. Voici un exemple à la figure 5.8 avec le thème `Warsaw` obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

 \begin{frame}
 \frametitle{Titre du frame}

 \begin{columns}

  \begin{column}{0.4\textwidth}
    \includegraphics[width=\textwidth]{hippopotenuse}
    \end{column}

    \begin{column}{0.6\textwidth}
    \begin{itemize}
    \item bla
    \item bla
    \end{itemize}
    \end{column}

 \end{columns}

 \end{frame}


\end{document}
```

5.6 Les images

Pour placer des images on utilisera `\includegraphics` (éventuellement dans un environnement `center` et/ou dans une colonne). Il est conseillé d'utiliser des dimensions relatives comme des pourcentage de

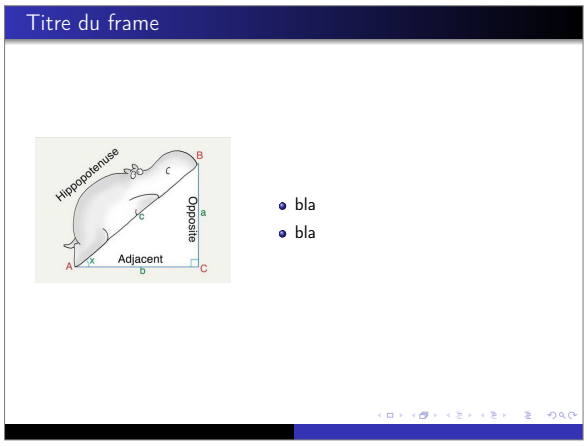


FIGURE 5.8 – Exemple de diapositives avec deux colonnes (thème `Warsaw`).

`\textwidth` ou `\paperwidth`. Étant donné qu'il ne s'agit pas de flottants, ça n'a aucun sens d'utiliser l'environnement `figure`.

5.7 La barre de navigation

`beamer` place dans le coin en bas à droite une barre de navigation qui permet de se déplacer dans la présentation avec la souris mais d'habitude on préfère naviguer avec le clavier ; pour enlever cette barre il suffit de mettre dans le préambule

```
\setbeamertemplate{navigation symbols}{}
```

Pour modifier la barre de navigation pour n'en garder que quelques boutons voir la documentation du package `beamer`.

Conseil : si le thème choisi ne montre pas le numéro de la diapositive, on pourra utiliser la place de la barre de navigation pour le faire apparaître :

```
\setbeamertemplate{navigation symbols}{\insertframenumber}
```

5.8 Mettre en avant des portions de texte

Pour souligner un ou des mots qui paraissent importants on peut toujours utiliser les commandes LaTeX `\emph{}` ou `\textbf{}` pour mettre en emphase ou en gras. Toutefois, `beamer` possède deux autres instructions mieux adaptés (et qui dépendent du thème et de la couleur du thème choisie) : les commandes `\structure{}` et `\alert{}` ainsi que les deux environnements du même type `\begin{structureenv}...\end{structureenv}` et `\begin{alertenv}...\end{alertenv}`.Voici un exemple à la figure 5.9 avec le thème `Warsaw` obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}
```

FIGURE 5.9 – Exemple de diapositives avec des mots mis en évidence (thème `Warsaw`).

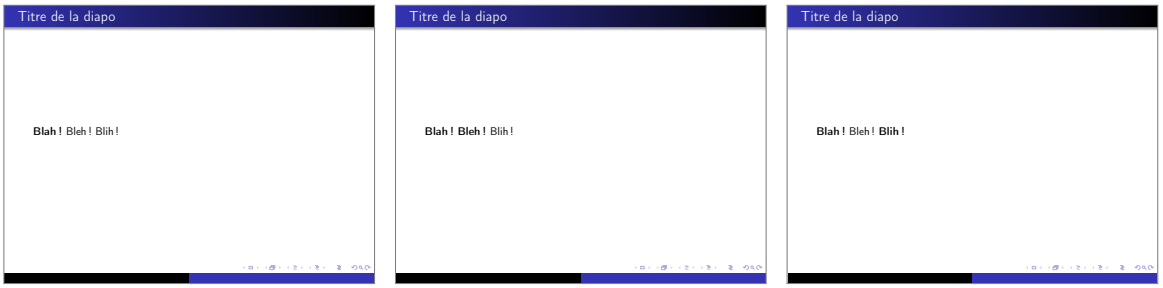
```
\begin{document}

 \begin{frame}
 \frametitle{Titre de la diapo}
 Bla bla bla \structure{bla bla bla bla} bla bla

 Bla bla \alert{bla bla} bla bla bla bla bla

 \begin{structureenv}
 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 \end{structureenv}

 \begin{alertenv}
 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
 \end{alertenv}

 \end{frame}

\end{document}
```

5.9 only, uncover, visible etc.

Un frame est constitué d'une suite d'une ou plusieurs layer qui peuvent être superposées. Certaines commandes supportent des spécifications de superposition : une spécification est le numéro (ou la liste des numéros ou une plage de numéros) des layers auxquelles la commande s'applique. Les numéros sont indiqués entre < et > après le nom de la commande.

Voici à la figure 5.10 un exemple du même frame avec trois layers obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}
```



(a) Layer 1. (b) Layer 2. (c) Layer 3.

FIGURE 5.10 – Exemple d'un même frame avec trois layers.

```
 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\textbf{Blah!}
\textbf<2>{Bleh!}
\textbf<3>{Blih!}
\end{frame}

\end{document}
```

Exemples des différents types de spécifications :

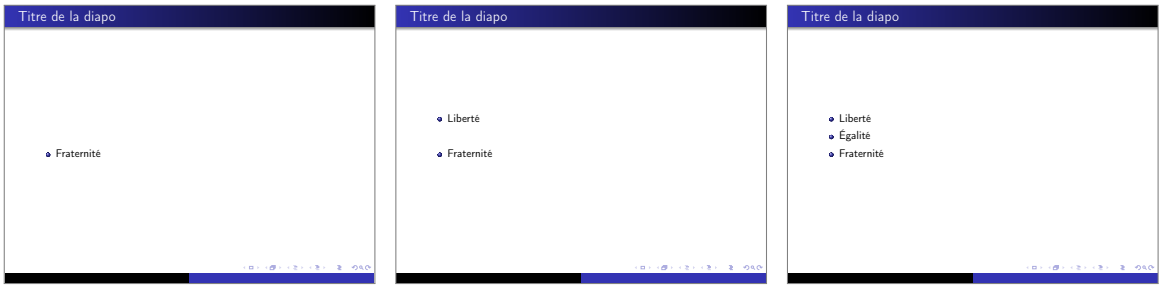
— <1> : seul le layer 1 du frame est concerné par la commande
— <1,2,4> : liste de layers du frame concernés par la commande
— <1-4> : plage de layers du frame concernés par la commande
— <3-> : les layers du frame concernés par la commande sont le 3e et tous les suivants
— <-3> : les layers du frame concernés par la commande sont tous ceux jusqu'au 3e inclus
— <-3,5,8-> : on peut combiner les écritures.

Commandes sujettes à spécifications :

— Commandes sur les polices : \textbf, \textit, \textsl, \textrm, \textsf
— Commande d'itération : \item
— Commandes de changement de couleur : \color, \alert, \structure
— Commandes et environnements propres à Beamer : \alert et \structure

Voici à la figure 5.11 un exemple du même frame avec trois layers obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
```



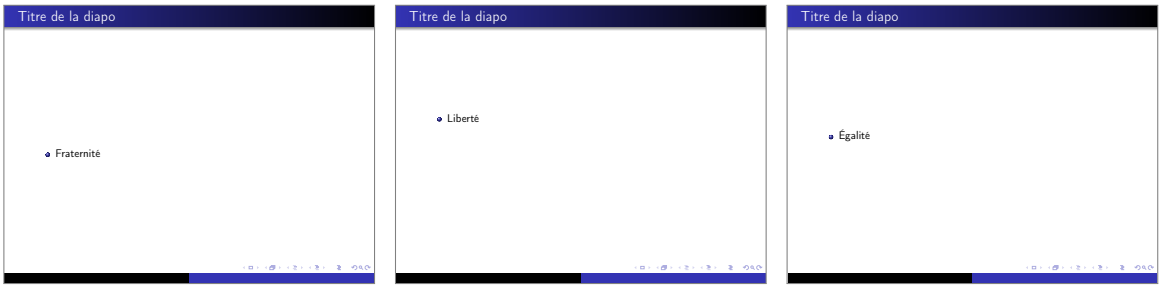
(a) Layer 1. (b) Layer 2. (c) Layer 3.

FIGURE 5.11 – Exemple d'un même frame avec trois layers.

```
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\begin{itemize}
\item<2-> Liberté
\item<3-> Égalité
\item<1-> Fraternité
\end{itemize}

\end{frame}

\end{document}
```

Voici à la figure 5.12 un exemple du même frame avec trois layers obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\begin{itemize}
\item<2> Liberté
\item<3> Égalité
\item<1> Fraternité
\end{itemize}
```



(a) Layer 1. (b) Layer 2. (c) Layer 3.

FIGURE 5.12 – Exemple d'un même frame avec trois layers.

```
\end{frame}

\end{document}
```

Commandes spécifiques :

— `\visible<plage>{texte}` : `texte` visible seulement sur `plage`

— `\uncover<plage>{texte}` : `texte` visible sur `plage` et en filigrane sur les autres layers

— `\invisible<plage>{texte}` : contraire de `\visible`

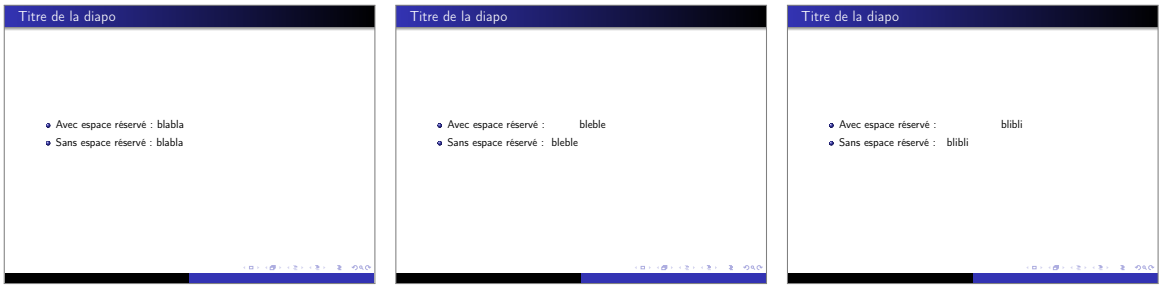
— `\only<plage>{texte}` : comme `\visible` mais ne réserve pas d'espace à ce qui est invisible

Voici à la figure 5.13 un exemple du même frame avec trois layers obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

 \usepackage{concrete}
 \usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc}   % sous linux
 \usepackage[T1]{fontenc}
 \usepackage[french]{babel}
 \usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\begin{itemize}
\item Avec espace réservé: \uncover<1>{blabla} \uncover<2>{bleble} \uncover<3>{
blibli}
\item Sans espace réservé: \only<1>{blabla}    \only<2>{bleble}    \only<3>{blibli}
\end{itemize}

\end{frame}

\end{document}
```

5.9.1 Pour aller plus loin

— La documentation du package `beamer` disponible à l'adresse https://www.ctan.org/pkg/beamer



(a) Layer 1. (b) Layer 2. (c) Layer 3.

FIGURE 5.13 – Exemple d'un même frame avec trois layers.

— Quelques conseils pour ne pas rater sa présentation orale (de N. Seguin) : http://seguin.perso.math.cnrs.fr/conseils.html

— et en anglais https://people.inf.ethz.ch/markusp/teaching/guides/guide-presentations-new.pdf

6 Compléments

6.1 Dessiner avec TikZ

Dans beaucoup d'ouvrages, les figures et les diagrammes souffrent de défauts rédhibitoires : traits trop épais, flèches dans un style complètement différent de celles du reste du document, polices de caractères différentes de celles du document, pixellisation, etc. Il y a plusieurs façon de faire des figures (mathématiques ou autre) avec LaTeX , et un des package les plus puissants est TikZ. Voici deux exemples aux figures 6.1 et 6.2.

Pour apprendre à utiliser ce très puissante package on pourra consulter :

— un manuel pour débuter http://math.et.info.free.fr/TikZ/

— le manuel complet (1165 pages) décrivant toutes les fonctions disponibles
http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf

— la galerie d'exemples en ligne http://www.texample.net/tikz/examples/all/

```
\begin{tikzpicture}[scale=1.7]
\shade[top color=blue,bottom color=gray!50]
      (0,0) parabola (1.5,2.25) |- (0,0);
\draw (1.05cm,2pt)
      node[above] {$\int_0^{3/2}x^2\mathrm{d}x$};
\draw[help lines] (0,0) grid (2.5,3.5)
     [step=0.25cm] (1,2) grid +(1,1);
\draw[->] (-0.2,0) -- (2.5,0) node[below] {$x$};
\draw[->] (0,-0.2) -- (0,3.5) node[left] {$f(x)$};
\foreach \x/\xtext in {1/1, 1.5/\frac{3}{2}, 2/2}
    \draw[shift={(\x,0)}] (0pt,2pt)
         -- (0pt,-2pt) node[below] {$\xtext$};
\foreach \y/\ytext in {1/1, 2/2, 2.25/\frac{5}{4},
3/3}
         \draw[shift={(0,\y)}] (2pt,0pt)
              -- (-2pt,0pt) node[left] {$\ytext$};
\draw (-.5,.25) parabola bend (0,0) (2,4)
    node[below right] {$x^2$};
\end{tikzpicture}
```

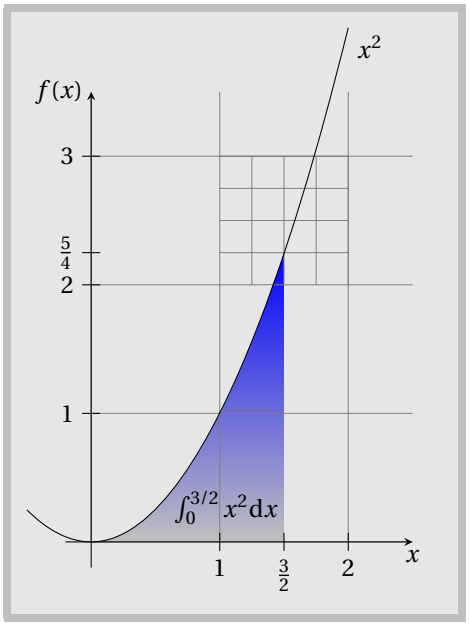


FIGURE 6.1 – Exemple de figures réalisée avec TikZ

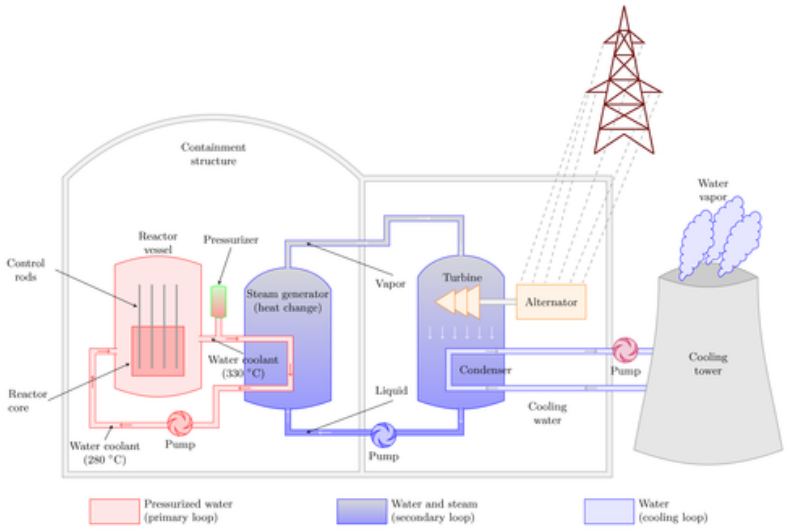


FIGURE 6.2 – https://texample.net/tikz/examples/pressurized-water-reactor/