



UNIVERSITÉ D'AVIGNON
ET DES PAYS DE VAUCLUSE

C E N T R E
D'ENSEIGNEMENT
ET DE RECHERCHE
EN INFORMATIQUE



Nom de la formation
Nom du parcours/spécialité
UE Nom de l'unité d'enseignement

Titre du document

Nom/numéro du groupe de travail

Prénom1 Nom1
Prénom2 Nom2
Prénom3 Nom3

CERI - LIA
339 chemin des Meinajariès
BP 1228
84911 AVIGNON Cedex 9
France

Tél. +33 (0)4 90 84 35 00
Fax +33 (0)4 90 84 35 01
<http://ceri.univ-avignon.fr>

Encadrement
Prénom4 Nom4
Prénom5 Nom5

Sommaire

Titre	1
Sommaire	2
1 Introduction	3
2 Compilation des documents	3
3 Généralités	4
3.1 Options de la classe	4
3.2 Informations du rapport	4
3.3 Résumé de la syntaxe	5
3.4 Mise en forme de base	6
3.5 Structure du document	7
3.5.1 Exemple de section de niveau 3	8
4 Éléments flottants et/ou numérotés	8
4.1 Figures	8
4.2 Tables	10
4.3 Algorithmes	13
4.4 Équations	15
4.5 Diagrammes de Gantt	17
5 Références	18
5.1 Références externes	18
5.2 Références croisées	18
5.3 Bibliographie	19
6 Conclusion	21
Bibliographie	22

1 Introduction

L'utilisation de logiciels tels que MS Word ou LibreOffice pour la rédaction de rapports aboutit généralement à des documents très hétérogènes, à la lisibilité douteuse, ce qui entraîne des difficultés lors de leur évaluation. Pour cette raison, c'est \LaTeX [7, 23] qui a été retenu pour les rapports de cette UE.

La mise en forme d'un document est aussi importante que son contenu. Il est clair qu'un rapport bien présenté mais dont le contenu est inadapté ou incorrect obtiendra une note faible. Mais de la même façon, un rapport dont le contenu est excellent et dont la présentation est mauvaise récoltera probablement une mauvaise note, car il sera difficile d'en comprendre les idées. Donc pour résumer, une bonne présentation est une condition nécessaire, mais pas suffisante pour produire un bon rapport. Tout ça pour dire que la mise en forme de vos rapports sera évaluée au même titre que leur contenu.

\LaTeX , pour simplifier, est un langage permettant de programmer la mise en forme d'un document. Donc, au lieu de composer votre document dans un traitement de texte, vous allez écrire seulement sa structure et son contenu sous la forme d'un code source (extension `.tex`). Vous devrez ensuite compiler ce code source pour obtenir un fichier `.pdf`.

\LaTeX et les outils associés sont libres (et gratuits). Il s'agit d'un standard dans le domaine de la recherche académique : la plupart des articles soumis à des conférences ou à des journaux scientifiques sont mis en forme grâce à ce système.

Mais le but principal de cette UE n'est pas d'apprendre à utiliser \LaTeX , c'est pourquoi le présent document tient lieu à la fois de tutoriel et de modèle pour la rédaction de rapports. Il ne s'agit pas d'un manuel sur \LaTeX : de nombreuses ressources disponibles sur le Web vous apprendront tout ce que vous devez savoir sur les bases de \LaTeX . Si vous n'avez jamais utilisé \LaTeX , il est recommandé de lire un texte d'introduction tel que ces WikiBooks en français [16] ou en anglais [17], en complément à ce document.

Le code source \LaTeX de ce document vous est fourni afin que vous l'utilisiez comme base pour vos propres rapports, après en avoir supprimé les parties inutiles et les commentaires.

2 Compilation des documents

Comme expliqué en introduction, \LaTeX permet de produire un document PDF en compilant un fichier source. On peut distinguer deux façons de réaliser cette compilation : soit via une application locale, soit grâce à une application Web.

La première méthode est beaucoup plus rapide lors de la compilation des documents, mais la procédure d'installation est plus ou moins complexe en fonction du système d'exploitation concerné. La seconde met plus de temps à produire le document, mais ne nécessite aucune installation, et intègre des fonctionnalités collaboratives (partage de documents, suivi des modifications, gestion des versions, système de commentaires, etc.).

Pour ces raisons, dans le contexte de cette UE, c'est cette dernière approche qui a été retenue¹. Nous utiliserons l'application Web *OverLeaf*, qui est disponible à l'adresse suivante :

<https://www.overleaf.com/signup?ref=d62cb1694be6>

Vous devez d'abord vous créer un compte (c'est gratuit), puis vous pourrez définir et compiler vos propres documents en ligne. Sélectionnez un identifiant correspondant à votre nom réel, et non pas à un surnom. Idéalement, quelque chose de la forme `prenom.nom`. En effet, la trace de votre activité sur OverLeaf est susceptible d'être utilisée lors de votre évaluation.

Le source \LaTeX du présent document est disponible sur OverLeaf à l'adresse suivante : <https://www.overleaf.com/latex/templates/modele-rapport-uapv/pdbgdpszgwr> Il s'agit d'un modèle à partir duquel vous devez instancier vos propres rapports. Pour cela, utilisez le bouton *Open as template* présent à l'URL indiquée ci-dessus.

1. Pour information, les anciennes versions de ce document (disponibles sur OverLeaf) décrivent la première approche.

3 Généralités

Cette section regroupe des informations basiques, dont certaines sont spécifiques à ce document \LaTeX en particulier (Sections 3.1 et 3.2), alors que les autres sont valables pour tous les documents \LaTeX en général (Sections 3.3 à 3.5).

3.1 Options de la classe

La première instruction d'un document \LaTeX est la classe du document produit (un peu l'équivalent d'une feuille de style). Pour ce rapport, il s'agit de la classe appelée `ceri`.

Celle-ci peut s'utiliser sans option, comme c'est le cas pour ce document :

```
\documentclass{ceri}
```

Il existe une option `light`, qui produit une version incomplète de la page de titre afin d'accélérer la compilation du document :

```
\documentclass[light]{ceri}
```

Remarque : ne rendez jamais un PDF produit avec cette option, elle est destinée à être utilisée seulement pendant la phase de rédaction.

L'option `full` rajoute quant à elle une table des figures et une table des tableaux en début de document :

```
\documentclass[full]{ceri}
```

À n'utiliser que pour de longs rapports contenant de nombreuses figures et/ou tableaux (ex. mémoire de M2). La page de titre apparaît aussi, comme quand on utilise la classe sans aucune option.

3.2 Informations du rapport

Les informations apparaissant sur la page de titre sont à indiquer en début de document, après la déclaration de la classe. Il faut utiliser pour cela des macro \LaTeX prédéfinies, qui sont décrites ci-dessous.

Formation. La commande `\major` est obligatoire. Elle sert à donner la formation délivrant l'UE :

```
\major{Master d'Informatique}
```

Parcours. La commande `\specialization` est obligatoire. Elle permet d'indiquer le parcours (ou la spécialité) de la formation :

```
\specialization{Ingénierie logicielle}
```

Unité d'enseignement. La commande `\course` est obligatoire. Elle est utilisée pour indiquer l'UE concernée par le rapport :

```
\course{Algorithmique et programmation}
```

Auteurs. La commande `\author` est elle aussi obligatoire. Elle sert à lister les auteurs, en les séparant par des `\\` :

```
\author{
  Sophie Dupont \\
  Abdel Cherif \\
  John McDouglas
}
```

Encadrants. La commande `\advisor` fonctionne exactement comme `\author`, excepté qu'elle est optionnelle. Elle permet de lister les enseignants encadrant le travail des étudiants. Attention, il ne faut l'utiliser que pour un rapport de stage, et non pas pour un projet classique.

```
\advisor{
  Fen Zhou \\
  Serigne Gueye
}
```

Groupe. La commande `\group` permet d'indiquer le nom ou le numéro du groupe de travail auquel les auteurs appartiennent. Elle est optionnelle, à n'utiliser que si le travail est réalisé en groupe :

```
\group{G12}
```

Titre. La commande `\title` est obligatoire. Elle reçoit en paramètre le titre du document :

```
\title{Rapport du premier semestre}
```

Résumé. La commande `\summary` est optionnelle. Elle est utilisée pour insérer un résumé en début de document.

```
\summary{Ce document décrit l'activité réalisée au premier semestre. Nous avons
  d'abord effectué une revue bibliographique, puis proposé un modèle de logiciel,
  et commencé l'implémentation.}
```

3.3 Résumé de la syntaxe

Un document \LaTeX contient principalement : du texte, qui est rendu tel quel ; et des commandes et environnements, qui déclenchent à des traitements spécifiques.

Commandes. Les noms de commande \LaTeX commencent par `\`, par exemple `\macommande`. Elles prennent éventuellement des options, entre crochets, par exemple `\macommande[monoption]`. Elles peuvent également recevoir des paramètres, entre accolades, par exemple `\macommande{monparametre}`. Ou les deux : `\macommande[monoption]{monparametre}`.

Environnements. Les environnements \LaTeX permettent de définir des traitements plus complexes. Ils sont délimités par deux commandes `\begin` et `\end` prenant le nom de l'environnement en paramètre. Par exemple, le texte du document est contenu dans l'environnement `document` :

```
\begin{document}
Texte du document, etc.
\end{document}
```

Commentaires. Les commentaires sont marqués grâce au caractère `%`, qui est valable pour toute une ligne (comme `//` en langage C ou Java). Par exemple :

```
% ceci est un commentaire
```

Le caractère `\` peut être utilisé pour désactiver ce caractère spécial (ainsi que tous les autres caractères spéciaux, d'ailleurs).

Paragraphes. Pour aller à la ligne et ainsi créer des paragraphes, il suffit de laisser une ligne vide. Par exemple :

```
Ce texte forme un premier paragraphe.
```

```
Et voici un second paragraphe.
```

Remarque : n'utilisez jamais `\\` pour aller à la ligne entre deux paragraphes.

3.4 Mise en forme de base

La mise en forme de base est décrite en détail dans les références déjà citées [16, 17]. Voici rapidement les commandes les plus utiles.

Gras et italique. Le texte peut être mis en **gras** avec la commande `\textbf` ou en *italique* avec `\textit` :

```
\textbf{Texte en gras}
\textit{Texte en italique}
```

Identificateurs. Tous les identificateurs (i.e. les noms de classes, méthodes, variables, constantes...) que vous citez dans le texte doivent être mis en forme avec la commande `\texttt`. Celle-ci permet d'insérer du texte avec une police de caractères de type listing (ex. *Courrier New* [18]). Par exemple, pour obtenir `MyClass`, on fait :

```
\texttt{MyClass}
```

Remarque : nous insistons particulièrement sur ce point, qui nous permet de rapidement repérer les identificateurs à la lecture de votre document, et donc de l'évaluer plus facilement.

Listes. Il est possible d'inclure des listes non-numérotées avec l'environnement `itemize` et la commande `\item`. Par exemple :

```
\begin{itemize}
  \item premier élément ;
  \item 2\ieme élément ;
  \item troisième élément.
  \begin{itemize}
    \item on peut aussi placer des sous-listes ;
    \begin{itemize}
      \item et rajouter encore d'autres niveaux ;
      \item etc.
    \end{itemize}
    \end{itemize}
  \item on continue au niveau supérieur ;
\end{itemize}
\item et au niveau encore supérieur.
```

Ce qui donne le résultat suivant :

- premier élément ;
- 2^e élément ;
- troisième élément.
 - on peut aussi placer des sous-listes ;
 - et rajouter encore d'autres niveaux ;
 - etc.
 - on continue au niveau supérieur ;
- et au niveau encore supérieur.

On peut également définir des listes numérotées automatiquement, en utilisant l'environnement `enumerate` (à la place d'`itemize`). On utilise toujours `\item` pour les éléments de la liste. Par exemple :

```
\begin{enumerate}
  \item premier élément ;
  \item 2\ieme élément ;
  \item troisième élément.
  \begin{enumerate}
    \item là encore, on peut placer des sous-listes ;
    \item ça marche exactement pareil que pour les listes non-numérotées ;
  \end{enumerate}
\end{enumerate}
```

```

\begin{itemize}
  \item on peut même mélanger les deux types de listes comme ici ;
  \item et ici.
\end{itemize}
\end{enumerate}
\end{enumerate}

```

Ce code source donne le résultat suivant :

1. premier élément ;
2. 2^e élément ;
3. troisième élément.
 - (a) là encore, on peut placer des sous-listes ;
 - (b) ça marche exactement pareil que pour les listes non-numérotées ;
 - on peut même mélanger les deux types de listes comme ici ;
 - et ici.

Couleurs. La commande `\textcolor` permet de modifier la couleur du texte lui-même, tandis que la commande `\colorbox` sert à changer sa couleur de fond. Par exemple, pour obtenir du **texte rouge** et un **fond vert**, on écrit :

```
du \textcolor{red}{texte rouge} et un \colorbox{green}{fond vert}
```

Pour les couleurs, vous pouvez soit utiliser le nom de la couleur en anglais, par exemple ici `red` ou `green`, soit définir une couleur précise en utilisant des valeurs numériques [16, 17].

Remarque : utilisez les couleurs avec une extrême parcimonie.

Notes de bas de page. Pour définir une note de bas de page, on utilise la commande `\footnote`, comme ici² :

```
comme ici\footnote{La numérotation se fait automatiquement.}
```

Remarque : évitez au maximum les notes de bas de page. Pour une meilleure lisibilité, placez vos remarques directement dans le texte, entre parenthèses.

Mode mathématique. Toutes les variables mathématiques doivent obligatoirement être mises en forme en utilisant la *mode mathématique*. On entre et on sort de ce mode en utilisant le caractère `$`. Par exemple, le fait d'écrire `$y = ax + b$` donne le rendu $y = ax + b$.

Certaines commandes telles que `\times` ne sont acceptées que dans le mode mathématique. Par exemple, `$7 \times 2 = 14$` donne $7 \times 2 = 14$. La Section 4.4 donne plus de détails sur l'utilisation du mode mathématique.

Divers. La commande `\ieme` permet d'insérer l'abréviation française pour *deuxième*, *troisième*, etc. : 2^e, 3^e...

Le caractère tilde `~` est un espace *insécable*, i.e. les deux mots qu'il relie ne peuvent pas être séparés par une fin de ligne.

3.5 Structure du document

Vous n'avez pas besoin de vous soucier de la mise en forme des titres de sections : celle-ci est déjà prévue dans la classe fournie. Vous devez seulement utiliser les commandes appropriées pour indiquer où commence chaque section. Celles-ci sont listées dans la Table 1.

Ainsi, pour définir une section de niveau 1, on fait :

```
\section{Titre de ma section}
```

```
\label{sec :MaSection}
```

Dans cette section, je vais expliquer bla bla bla bla bla...

2. La numérotation se fait automatiquement.

Niveau hiérarchique	Commande correspondante
1	<code>\section</code>
2	<code>\subsection</code>
3	<code>\subsubsection</code>
4	<code>\paragraph</code>

Table 1. Commandes utilisées pour définir les titres de sections et sous-sections.

Chaque section est identifiée de façon unique grâce à un label. Celui-ci doit être spécifié juste après la définition de la section, en utilisant la commande `\label`. Celle-ci reçoit le label en paramètre (dans l'exemple ci-dessus : `sec :MaSection`). Quel que soit le niveau de la section, le label qui lui est associé doit être préfixé par `sec :`.

Le niveau 3 n'est pas réellement utilisé dans ce document, car il n'y est pas nécessaire. Pour l'exemple, le voici :

3.5.1 Exemple de section de niveau 3

Notez qu'il est nécessaire de mettre une section de niveau 3 entre la section de niveau 2 et celle de niveau 4, sinon une section factice de niveau 3 numérotée 0 sera automatiquement créée.

Le niveau 4 n'est pas vraiment une section mais plutôt un titre de paragraphe, comme ci-dessous :

Exemple de section de niveau 4. Pour cette raison, le titre n'est pas sur une ligne séparée, mais il est directement placé dans le paragraphe concerné.

Remarques : ne définissez pas une section seulement pour y placer 1 ou 2 lignes de texte. Une section doit avoir un contenu assez long, sinon ce n'est pas la peine de la créer.

4 Éléments flottants et/ou numérotés

\LaTeX permet d'insérer un certain nombre d'éléments dits *flottant* : figures (Section 4.1), tables (Section 4.2), et algorithmes (Section 4.3). On les qualifie de *flottant* car \LaTeX est susceptible de ne pas les insérer exactement là où l'utilisateur les définit. Ces éléments sont numérotés automatiquement par \LaTeX . Il en va de même pour les équations (Section 4.4), bien qu'il ne s'agisse pas d'éléments flottant.

Ces éléments doivent toujours posséder une légende (sauf les équations) et être numérotés. De plus, ils doivent être mentionnés, décrits et commentés dans le texte. Toute référence à l'un de ces éléments doit être faite en utilisant la commande `\ref`, décrite plus en détail dans la Section 5.2.

4.1 Figures

On insère une figure en utilisant l'environnement `figure` :

```
\begin{figure}[htb !]
  \centering
  \includegraphics[scale=0.5]{images/univ.jpg}
  \caption[Université d'Avignon]{La façade de l'Université d'Avignon.}
  \label{fig :uapv}
\end{figure}
```

Ce code source a pour résultat la Figure 1.

Positionnement. L'option `[htb !]` sert à positionner la figure. Notez que \LaTeX se charge de placer automatiquement les éléments flottant dans le document.

Il est possible que l'élément ne soit pas placé exactement à l'endroit spécifié par l'utilisateur, si le contenu du document l'empêche. C'est normal, laissez le système gérer cela. C'est d'ailleurs pour



Figure 1. La façade de l'Université d'Avignon.

cette raison qu'il faut systématiquement mentionner les éléments flottant en utilisant le système de références croisées de \LaTeX (cf. Section 5.2).

La commande `\centering` permet de centrer la figure, et doit toujours être utilisée.

Contenu. La figure est ici une image contenue dans le fichier, qui est insérée grâce à la commande `\includegraphics`. Celle-ci prend en paramètre le chemin et le nom du fichier concerné, qui est ici `images/univ.jpg`.

L'option `scale=x` permet de spécifier le niveau de zoom x de l'image avec `scale=x` (taille divisée par 2, dans l'exemple). Il est aussi possible de préciser sa largeur avec `width=x` ou sa hauteur avec `height=x`. La valeur x peut alors être exprimée en cm (ex. `width=15cm`), ou bien comme une fraction de la largeur de la ligne grâce à la commande `\textwidth` (ex. `width=0.75\textwidth`).

Légende. Toute figure doit obligatoirement être décrite par une légende. Celle-ci est insérée grâce à la commande `\caption`. Elle prend en paramètre le texte de la légende (ici : *La façade de l'Université d'Avignon*), et éventuellement en option une version plus courte de cette légende (ici : *Université d'Avignon*). Cette version courte, le cas échéant, est destinée à être affichée dans la *table des figures* (cf. Section 3.1). \LaTeX se charge de numérotter automatiquement la figure.

Label. Chaque figure doit être désignée par un label unique dans le document, afin d'y faire des références croisées. Ce label est défini juste après la légende, en utilisant la commande `\label` (qui prend le label en paramètre). Le nom unique de la figure est préfixé par `fig :`. Dans l'exemple, le label est `fig :uapv`.

Diagrammes. Pour vos diagrammes, produisez des images *vectérielles* [22], au format PDF. Cela permet au lecteur de zoomer sans perte de qualité. Attention à ne pas enregistrer simplement une image *matricielle* [21] dans un PDF : cela n'aurait aucun intérêt. De nombreux logiciels gratuits permettent d'éditer des diagrammes et de les enregistrer sous forme vectorielle.

Par exemple, le logiciel libre Dia^{3,4} est disponible pour la plupart des plateformes. En plus du format PDF, il permet même de générer directement du code \LaTeX , qui peut ensuite être intégré tel

3. <http://dia-installer.de>

4. On ne met d'URL dans le document que pour indiquer un lien vers une ressource non-bibliographique, comme ici (un logiciel).

quel dans le document. Pour produire ce type de code, suffit d'aller dans *Fichier > Exporter*, et de choisir le format *Macros PGF LaTeX (*.tex)*. Pour insérer une telle figure dans le texte, la commande est légèrement différente de celle présentée auparavant :

```
\begin{figure}[htb !]
  \centering
  \resizebox{0.75\linewidth}{!}{\input{images/classdiag.tex}}
  \caption{Exemple de diagramme.}
  \label{fig :diag}
\end{figure}
```

Ce code source a pour résultat la Figure 2, qui affiche le fichier `images/classdiag.tex`. Notez que la valeur 0.75 peut être modifiée : elle signifie que la figure occupe 75% de la largeur de la page.

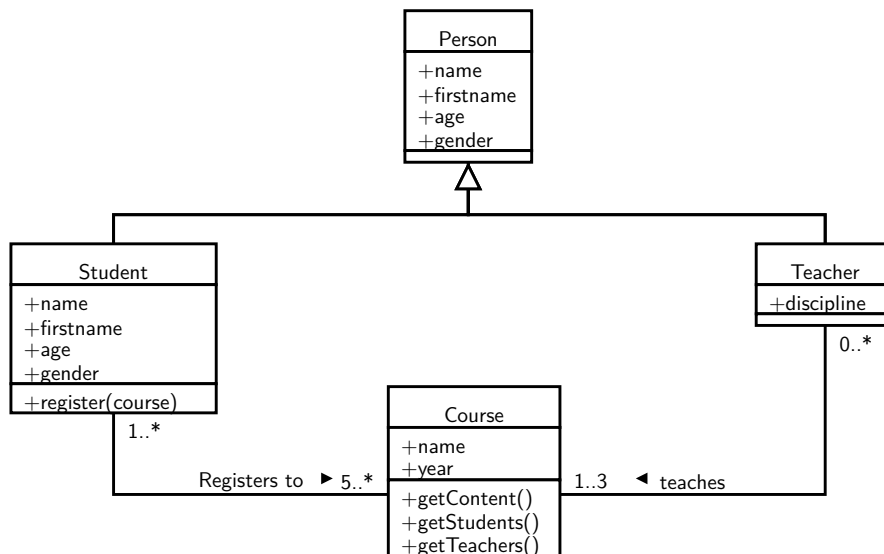


Figure 2. Exemple de diagramme.

4.2 Tables

L'environnement permettant d'insérer une table est `table` :

```
\begin{table}[htb !]
  \centering
  \rowcolors{1}{LightColor}{}
  \begin{tabular}{l c r r}
    \hline
    \rowcolor{DarkColor}
    \textbf{Texte} & \textbf{Texte} & \textbf{Entiers} & \textbf{Réels} \\
    \hline
    Blabla bla bla bla bla & Blabla bbla bla & 21 & 12,62 \\
    Blabla bla bla bla bla & Blabla bla bla bla bla & 12~356 & 3~456,21 \\
    Blabla bla bla & Blabla bla bbla bla & 1 & 45,87 \\
    Blabla blabla bla bla & $y = ax + b$ & 456 & 72,16 \\
    Blabla bla blaa bla & Blabla bla bla bla & 78 & 89,28 \\
    \hline
  \end{tabular}
  \caption[Exemple de table hétérogène]{Exemple de table contenant du texte, des
    valeurs entières, réelles et des formules mathématiques.}
  \label{tab :exemple}
\end{table}
```

Le résultat du code source ci-dessus est la Table 2.

Texte	Texte	Entiers	Réels
Blabla bla bla bla bla	Blabla bbla bla	21	12,62
Blabla bla bla bla	Blabla bla bla bla bla	12 356	3 456,21
Blabla bla bla	Blabla bla bbla bla	1	45,87
Blabla blabla bla bla	$y = ax + b$	456	72,16
Blabla bla blaa bla	Blabla bla bla bla	78	89,28

Table 2. Exemple de table contenant du texte, des valeurs entières, réelles et des formules mathématiques.

Positionnement. Le contrôle de la position de la table au moyen de l'option `[htb !]` de l'environnement `table`, ainsi que de la commande `\centering`, fonctionne exactement comme pour les figures.

Contenu. La syntaxe permettant de définir le contenu des tables en \LaTeX est relativement compliquée. Vous vous limiterez à des tables de la forme décrite dans cette sous-section : cette mise en forme doit être respectée, afin de garder des tables lisibles.

Le contenu de la table est défini au moyen de l'environnement `tabular`. Le paramètre permet de préciser à la fois le nombre de colonnes contenues dans la table, et leur alignement :

- `l` (*left*) pour une colonne alignée à gauche ;
- `r` (*right*) pour qu'elle soit alignée à droite ;
- `c` (*center*) pour qu'elle soit centrée.

Les données de la table sont ensuite décrites ligne par ligne. Chaque ligne de la table est décrite par une ligne de texte terminée par `\\`. Sur la ligne de texte, les colonnes sont séparées par le caractère `&`.

Mise en forme. Les lignes horizontales de séparation sont tracées grâce à la commande `\hline`. Une ligne sur deux est grisée afin de faciliter la lecture, grâce à la commande `\rowcolors{1}{LightColor}{}` (placée avant `tabular`).

La première ligne de la table contient les titres des colonnes. Le texte contenu dans cette ligne de titre doit être en **gras** (commande `\textbf`). La ligne doit être colorée de façon plus sombre, grâce à la commande `\rowcolor{DarkColor}` (placée juste avant le début de la ligne).

Vos tables doivent ressembler exactement à celles données en exemple dans ce document : couleurs, séparations, légende située en-dessous, etc. Les valeurs numériques doivent être alignées à droite. Les réels doivent être définis de manière à avoir le même nombre de décimales. Les valeurs textuelles peuvent être alignées à droite ou à gauche, ou bien centrées. Il est possible d'insérer des équations en-ligne (ie. non-numérotées) dans une table, en utilisant simplement le mode mathématique.

Légende. La légende de la table est insérée exactement comme pour les figures, en utilisant la commande `\caption`.

Label. Le label permettant de faire référence à la table s'insère aussi comme pour les figures, juste après la légende, et grâce à la commande `\label`. Il y a toutefois une différence : le label associé à un table est préfixé par `tab` : (au lieu de `fig` : pour les figures).

Table haute. Si votre table est haute et étroite, alors vous devez la décomposer afin de mieux utiliser l'espace. Ceci est illustré avec la Table 3, qui contient 50 lignes pour seulement 2 colonnes.

La décomposition en plusieurs parties est réalisée en utilisant l'environnement `minipage` (dans `table`). Voici un aperçu du code source de la Table 3 (consultez le code source de ce document pour le voir en entier) :

Col1	Col2	Col1	Col2	Col1	Col2	Col1	Col2	Col1	Col2
1	hdc	11	ytr	21	ioe	31	zyc	41	oze
2	orf	12	xcv	22	snu	32	opv	42	anb
3	seh	13	fsd	23	dgp	33	azv	43	kir
4	lig	14	pui	24	zbu	34	xdf	44	sfk
5	jhg	15	gsf	25	zvz	35	yzd	45	bed
6	azr	16	aze	26	orv	36	sgg	46	pyj
7	opi	17	bcv	27	czr	37	sss	47	zdz
8	hgj	18	wxc	28	azh	38	zdv	48	pyj
9	xvc	19	uyr	29	cyh	39	sgu	49	zen
10	ree	20	ndz	30	ytj	40	vzd	50	oef

Table 3. Exemple de table haute et étroite, décomposée en 5 parties pour occuper plus efficacement l'espace disponible.

```

\begin{table}[htb !]
  \rowcolors{1}{LightColor}{}
  \begin{minipage}[t]{0.19\linewidth} % 1ère partie de la page...
    \centering
    \begin{tabular}{l l}
      \hline
      \rowcolor{DarkColor} \textbf{Col1} & \textbf{Col2} \\
      \hline
      1 & hdc \\
      2 & orf \\
      ...
      \hline
    \end{tabular}
  \end{minipage}
  \begin{minipage}[t]{0.19\linewidth} % 2ème partie de la table...
    \centering
    \begin{tabular}{l l}
      \hline
      \rowcolor{DarkColor} \textbf{Col1} & \textbf{Col2} \\
      \hline
      11 & ytr \\
      12 & xcv \\
      ...
      \hline
    \end{tabular}
  \end{minipage}
  \begin{minipage}[t]{0.19\linewidth} % 3ème partie de la table...
    \centering
    \begin{tabular}{l l}
      \hline
      \rowcolor{DarkColor} \textbf{Col1} & \textbf{Col2} \\
      \hline
      21 & ioe \\
      22 & snu \\
      ...
      \hline
    \end{tabular}
  \end{minipage}
  ...
\end{table}

```

Chaque partie de la table doit être placée dans un environnement minipage. Ici, le paramètre

0.19\linewidth indique que la chaque partie de la table occupe 19% de la largeur de la page. Il faut donc adapter cette valeur en fonction du nombre de parties à afficher, et de leurs largeurs respectives.

Fichier externe. Il est possible de lire le contenu de la table dynamiquement dans un fichier externe. Par exemple, la Table 4 est obtenue à partir du fichier `donnees/test.csv`. Voyez la documentation du package `csvsimple`⁵ pour plus de détails.

Nom	Prénom	Genre	Âge
Karamazov	Fiodor	H	54
Karamazov	Dimitri	H	28
Karamazov	Ivan	H	24
Karamazov	Alexeï	H	19
Smerdiakov	Pavel	H	N/A
Svietlova	Agrafena	F	22
Verkhovtseva	Katerina	F	N/A

Table 4. Exemple de table obtenue automatiquement à partir du fichier CSV `donnees/test.csv`.

Le code source suivant est un aperçu de celui de la Table 4. Notez que l'on définit manuellement l'en-tête de la table. On indique le nom du fichier CSV (ici : `donnees/test.csv`) ainsi que les colonnes de ce fichier que l'on veut insérer dans la table (ici : `nom`, `prenom`, `genre`, `age`). On n'est pas obligé d'insérer toutes les colonnes du fichier, ni de respecter leur ordre dans le fichier.

```
\begin{table}[htb !]
...
\begin{tabular}{l l l r}
\hline
\rowcolor{DarkColor}
\textbf{Nom} & \textbf{Prénom} & \textbf{Genre} & \textbf{Âge} \\
\hline
\csvreader[head to column names, late after line=\\]
% on indique le nom du fichier CSV ici :
{donnees/test.csv}{
% on indique les noms de colonnes à insérer dans la table :
% (cf. le fichier CSV pour voir les correspondances)
{\nom & \prenom & \genre & \age}
\hline
\end{tabular}
...
\end{table}
```

4.3 Algorithmes

Un algorithme peut être décrit sous la forme d'un diagramme de type *flowchart* [20], et apparaîtra alors dans le rapport en tant que *figure*. On peut aussi donner l'algorithme sous la forme de pseudo-code. Dans ce cas là, on utilise l'environnement `algorithm` :

```
\begin{algorithm}[htb !]
\KwData{$x, y$}
\KwResult{M}

\BlankLine
\tcp{Initialisation}
$z \leftarrow 50$;
\end{algorithm}
```

5. <https://ctan.org/pkg/csvsimple>

```

\BlankLine
\tcp{Boucle principale}
\For{$i \leftarrow 1$ to $x$}{
  \While{$y \leq 18$}{
    \For{$u \in V$}{
      \If{$u \neq v$}{
        $a \leftarrow a + 1$; \label{line :example}
        $y \leftarrow x + y / 2$;
      }
      \eIf{$x > y$}{
        $x \leftarrow x \times y$;
      }{
        $y \leftarrow \sqrt{x + y / 2}$;
      }
    }
  }
}

\BlankLine
\tcp{Un autre commentaire}
  $B \leftarrow (x + y)^a$;
}
$M \leftarrow ||B||$;

\caption{Exemple d'algorithme bidon, représenté sous forme de pseudo-code.}
\label{algo :exemple}
\end{algorithm}

```

Le résultat de ce code source est affiché dans l'Algorithme 1.

```

Data :  $x, y$ 
Result :  $M$ 

// Initialisation
1  $z \leftarrow 50$ ;

// Boucle principale
2 for  $i \leftarrow 1$  to  $x$  do
3   while  $y \leq 18$  do
4     for  $u \in V$  do
5       if  $u \neq v$  then
6          $a \leftarrow a + 1$ ;
7          $y \leftarrow x + y/2$ ;
8       end
9       if  $x > y$  then
10         $x \leftarrow x \times y$ ;
11      else
12         $y \leftarrow \sqrt{x + y/2}$ ;
13      end
14    end
15  end

// Un autre commentaire
16  $B \leftarrow (x + y)^a$ ;
17 end
18  $M \leftarrow ||B||$ ;

```

Algorithme 1 : Exemple d'algorithme bidon, représenté sous forme de pseudo-code.

Positionnement. La position de l'algorithme est spécifiée via l'option `[htb !]` de l'environnement, comme pour les figures et les tables. Par contre, à la différence des figures et des tables, il ne faut pas centrer l'algorithme (donc : pas de commande `\centering` ici).

Contenu. Les commandes `\KwData` et `\KwResult` permettent respectivement de définir les entrées et sorties de l'algorithme (notées `Data` et `Result` dans l'Algorithme 1).

La commande `\BlankLine` permet de sauter une ligne dans l'algorithme. La commande `\tcp` prend du texte en paramètre, et l'affiche sous forme de commentaire dans l'algorithme. La fin d'une ligne comportant une simple instruction ou affectation doit être marquée par `\;`.

Les commandes `\For` et `\While` permettent de définir respectivement des boucles *Pour* et *Tant que*. Elles prennent chacune deux paramètres : le premier pour définir la condition de la boucle, et le second pour définir les instructions qui la constituent.

Les commandes `\If` et `\eIf` permettent de définir respectivement un *Si* simple (ie. sans *Sinon*) et un *Si...Sinon*. La première prend 2 paramètres, et la seconde 3. Le premier paramètre est la condition du *Si*, le deuxième contient les instructions exécutées si cette condition est vérifiée, et le troisième (seulement pour `\eIf`) celles exécutées dans le cas contraire (c'est le *Sinon*).

Notations. Toutes les variables manipulées doivent être mises en forme en utilisant le mode mathématique (cf. Sections 3.4 et 4.4). En particulier, les affectations doivent être notées avec le symbole \leftarrow (commande `\leftarrow` utilisée en mode mathématique).

Vous devez utiliser au maximum des notations *mathématiques* (par opposition à *informatiques*) dans votre pseudo-code, car cela permet de le rendre plus compact et plus lisible. Par exemple, si on veut parcourir tous les éléments d'une liste L avec une variable e , on utilisera en condition de la boucle *Pour* une expression de la forme : **for** $e \in L$ **do**... (commande `\in` pour \in).

Choisissez des noms de variables d'une seule lettre (possiblement une lettre grecque) et non pas des expressions à rallonge comme vous le feriez dans un programme informatique. Vous pouvez par contre utiliser des noms longs pour les fonctions, pour des raisons d'intelligibilité.

Légende. La légende de l'algorithme est insérée exactement comme pour les figures et tables, en utilisant la commande `\caption`.

Label. Le label permettant de faire référence à l'algorithme s'insère aussi comme pour les figures et tables, juste après la légende, et grâce à la commande `\label`. Ce label diffère par son préfixe, qui est `algo` : (au lieu de `fig` : pour les figures et `tab` : pour les tables).

Il est aussi possible de définir un label pour certaines lignes en particulier, comme par exemple la ligne 6 dans l'Algorithme 1. On préfixe alors le label par `line` :, comme dans l'exemple.

Description informelle. Le pseudo-code constitue une description *formelle* de l'algorithme. En plus de cela, il vous faut en donner une description *informelle*, c'est-à-dire expliquer avec du texte comment il fonctionne. Donnez d'abord le principe général, puis entrez dans les détails. Listez et décrivez aussi chaque variable utilisée dans l'algorithme. Ajoutez ensuite un exemple d'application sur des données simples, en déroulant l'algorithme à la main et en expliquant les différentes étapes du traitement.

4.4 Équations

Le système \LaTeX est particulièrement bien adapté à la mise en forme de formules mathématiques. Les équations peuvent y être introduites sous deux formes : *en-ligne* et *hors-ligne*.

Une équation en-ligne est définie en utilisant le mode mathématique (cf. Section 3.4), par exemple $y = ax + b$. Ce texte s'intègre à un paragraphe normal, ou même à une table. Cependant, la mise en forme en-ligne n'est pas adaptée à des formules prenant beaucoup de place. De plus, elle ne permet

pas de numéroter l'équation, et donc d'y faire référence plus tard. Pour résoudre ces deux problèmes, il faut utiliser la mise en forme hors-ligne.

L'insertion d'une équation numérotée, en mode hors-ligne, se fait grâce à l'environnement `equation` :

```
\begin{equation}
  f(s) = h(s) + \max(g_1(s), g_2(s))
  \label{eq :exemple}
\end{equation}
```

L'équation correspondant à ce code source prend la forme suivante :

$$f(s) = h(s) + \max(g_1(s), g_2(s)) \quad (1)$$

Comme on peut le voir ci-dessus, *hors-ligne* signifie que l'équation est clairement séparée du texte.

Positionnement et légende. À la différence des figures, tables, et algorithmes, une équation n'est pas un élément flottant, donc elle est insérée à l'endroit où l'utilisateur la définit. Par conséquent, il n'y a pas lieu d'utiliser d'option de positionnement (donc pas de `[htb !]`) ni de définir de légende (donc pas de `\caption`). De plus, elle est centrée automatiquement (donc pas de `\centering`).

Mise en forme. Les instructions indiquées ici sont également valables pour la mise en forme des équations en-ligne. \LaTeX offre de nombreuses possibilités de mise en forme mathématique, voici les principales. Voyez [16, 17] pour plus de détails.

L'opérateur de multiplication s'obtient avec la commande `\times` (ex. $x \times y$) : interdit d'utiliser l'astérisque `*`. Les mises en indice et en exposant s'obtiennent respectivement avec `_` (*underscore*) et `^`, par exemple : $\Phi = x^2 + x^{a+b} - x_1 - x_{n-1}$. La plupart des fonctions remarquables ont une commande dédiée, par exemple les fonctions trigonométriques (ex. $\cos x$), le logarithme (ex. $\ln x$), minimum et maximum (ex. $\min x$), etc. L'opérateur de sommation s'obtient avec la commande `\sum`, par exemple $\sum_{i=1}^{10} x + y$.

Les matrices ne doivent *pas* être représentées par des tables : le mode mathématique est prévu à cet effet, comme illustré par l'Eq.(2) :

$$M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \quad (2)$$

Vous devez utiliser l'environnement `pmatrix`, qui fonctionne à peu près comme `tabular` :

```
\begin{equation}
  M =
  \begin{pmatrix}
    a & b & c \\
    d & e & f \\
    g & h & i
  \end{pmatrix}
  \label{eq :matrice}
\end{equation}
```

Label. Le label d'une équation est spécifié comme pour les figures, tables, et algorithmes : en utilisant la commande `\label`. À noter qu'il faut utiliser le préfixe `eq` : pour définir ce label.

Notations. Vos variables et fonctions doivent avoir des noms courts : une lettre et un indice, maximum. Pas de nom du type *mafonction*, mais plutôt f , f' , f_1 , etc.

Ne mélangez pas les notations informatiques et mathématiques. Par exemple, n'utilisez pas la notation tableau (`array`) dans une équation : pas de $f = a + \text{tab}[12]$.

4.5 Diagrammes de Gantt

Ce thème est configuré pour permettre l'insertion de diagrammes de Gantt basiques tel que celui présenté dans la Figure 3. Si vous trouvez mieux ailleurs, ou si vous avez besoin de plus de fonctionnalités, vous pouvez utiliser un autre outil, y compris externe. Cependant, veuillez alors à ce que cet outil génère bien un fichier PDF. Notez que l'avantage de l'approche présentée ci-dessous est que le diagramme est facilement éditable directement dans le document \LaTeX .

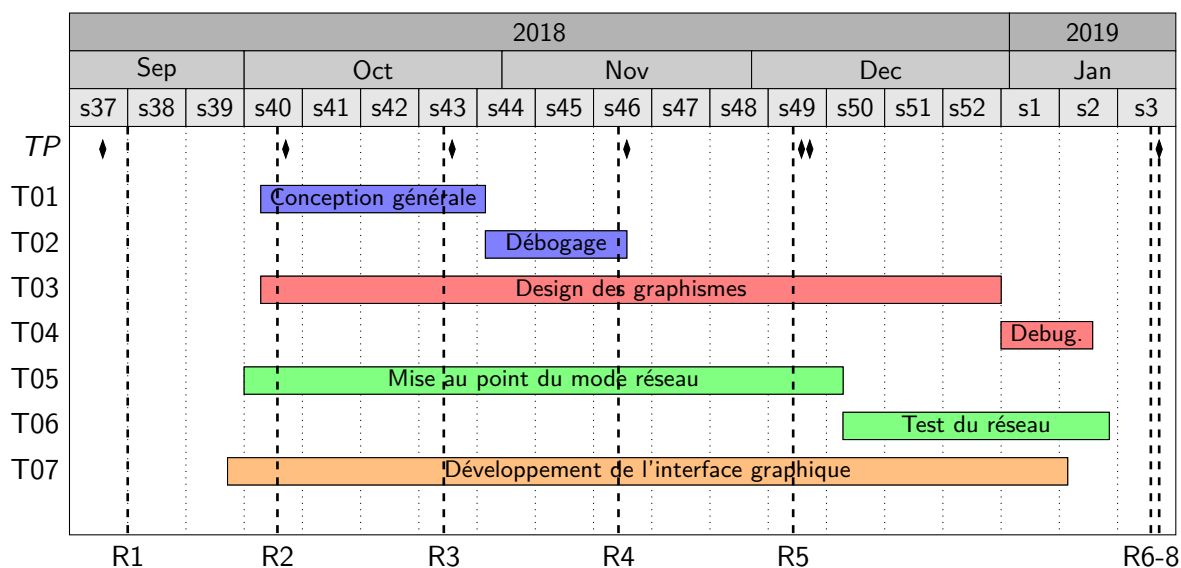


Figure 3. Diagramme de Gantt pour le premier semestre. Chaque couleur correspond à un étudiant : **Prénom1 Nom1**, **Prénom2 Nom2**, **Prénom3 Nom3**, et **Prénom4 Nom4**. Les \blacklozenge représentent les séances de TP, tandis que les lignes verticales correspondent aux dates de rendu.

Vous devez utiliser l'environnement `gantchart`, lui même inséré dans un environnement `figure`. L'extrait de code source ci-dessous présente les points principaux :

```
\begin{figure}[htb !]
...
\setcounter{myWeekNum}{37} % semaine de départ
\begin{gantchart}[...
  x unit=0.11cm, % largeur d'une journée
  ...
  ]{2018-09-10} % date de début de la période considérée
  {2019-01-20} % date de fin de la période considérée
  % indications temporelles
\gantttitlecalendar[title/.style={fill=black !30,draw=black}]{year} \\
...
% séances de TP
\ganttmilestone{TP}{2018-09-13} %TP01
...
% tâches
\ganttbarbis{T01}{Tâche 01}{2018-10-03}{2018-10-29}{blue !50} \\
...
% dates de rendu
\drawverticalline{2018-09-16}{R1}
...
\end{gantchart}
...
\end{figure}
```

Période représentée. La commande `\setcounter{myWeekNum}{xx}` permet d'indiquer le numéro `xx` (dans l'année) de la semaine de départ pour la période représentée.

Les deux derniers paramètres de l'environnement `ganttchart` doivent être utilisés pour indiquer le début et la fin de la période considérée. Dans l'exemple, il s'agit de 2018-09-10 et 2019-01-20.

Jalons. Il est possible de définir des jalons en utilisant la commande `\ganttmilestone`. Elle prend en paramètre un texte optionnel (affiché dans la marge gauche du diagramme) et la date associée au jalon.

Ces jalons prennent la forme de losanges noirs dans le diagramme. Dans l'exemple, on les utilise pour représenter les séances de TP.

Tâches. Les tâches sont ajoutées au moyen de la commande `\ganttbarbis`, et prennent la forme de barres horizontales dans le diagramme. La commande prend en paramètre un nom court (affiché dans la marge gauche), un nom plus long (affiché dans la barre représentant la tâche) et une couleur (utilisée pour colorer la barre). Le nom long doit être ajusté pour tenir dans la largeur de la barre.

Échéances. Les échéances apparaissent sous forme de droites verticales, obtenues au moyen de la commande `\drawverticalline`. Celle-ci prend en paramètre une date permettant de situer l'échéance, et un nom court qui est placé au-dessous de la droite dans le diagramme.

5 Références

On distingue trois types de références dans un document \LaTeX . Les références externes (Section 5.1) correspondent à des liens vers des documents Web. Les références croisées (Section 5.2) sont des liens vers des éléments numérotés du document (figures, tables, algorithmes, équations). Enfin, les références bibliographiques (Section 5.3) pointent vers des documents listés dans la section bibliographique.

5.1 Références externes

On crée un hyperlien au moyen de la commande `\url`, qui prend l'URL en paramètre. Par exemple :

```
\url{http://univ-avignon.fr}
```

Ce qui donne le rendu suivant : <http://univ-avignon.fr>.

Indiquer directement un hyperlien de cette façon n'est approprié que s'il s'agit d'une ressource qui n'est pas un document (par exemple un logiciel, ou des données), et si elle n'est citée qu'une seule fois dans tous le document. Dans le cas contraire, c'est à dire si la ressource est un document ou si elle est citée plusieurs fois dans le document, vous devez définir une référence bibliographique (cf. Section 5.3).

5.2 Références croisées

On a vu précédemment que \LaTeX permet de définir un label pour marquer un endroit particulier du document, grâce à la commande `\label`.

Il est ensuite possible de faire référence à ce label, en utilisant la commande `\ref`, qui prend en paramètre le nom du label concerné. Par exemple, pour faire référence au label `abcdef`, on utilisera la commande suivante :

```
\ref{abcdef}
```

La référence apparaît dans le texte sous la forme d'un lien hypertexte, ce qui est pratique pour naviguer dans le document. Si le label correspond à un objet numéroté (section, figure, table,

algorithme, équation...), alors c'est son numéro qui sera affiché. La dernière colonne de la Table 5 illustre cela.

Type d'élément	Préfixe du label	Exemple de référence
Section	sec :	Section 5
Figure	fig :	Figure 1
Tableau	tab :	Table 5
Équation	eq :	Eq.(2)
Algorithme	algo :	Algorithme 1
Ligne dans un algorithme	line :	Ligne 6

Table 5. Préfixes à utilisés pour définir les noms des labels, et exemples de rendus.

La convention veut qu'on utilise des préfixes spécifiques dans les noms des labels en fonction du type d'objet référencé. Vous devez utiliser cette convention dans vos rapports (voir la Table 5). La deuxième colonne de la Table 5 rappelle ces conventions.

Il est absolument obligatoire que tout élément flottant (figure, table, algorithme) soit cité explicitement dans votre texte. Cette citation doit se faire nécessairement par `\ref`. Donc chacun doit avoir son propre label. De plus, la nature de l'élément concerné doit être explicitement mentionné, comme indiqué dans la dernière colonne de la Table 5. Autrement dit, il ne faut pas indiquer simplement "5", mais bien "Table 5".

Bien sûr vous ne devez pas vous contenter de citer chaque élément flottant : il doit être suffisamment décrit, et de façon précise. Un élément qui n'est pas mentionné dans le texte ne sert à rien, et on le considérera comme du remplissage lors de l'évaluation de votre rapport.

5.3 Bibliographie

Toutes les informations que vous intégrez à votre rapport et/ou que vous avez utilisées lors de votre travail, mais qui n'ont pas été produites par vous-même, doivent être référencées *explicitement*. Vous devez donc indiquer tout document ou ressource que vous utilisez, y compris ce qui est récupéré sur internet. Cela vaut pour tout type d'information : document texte, images, vidéos, code source, algorithmes, idées diverses.

Autrement dit, vous avez le droit de réutiliser absolument tout ce que vous désirez. Par contre, vous devez à chaque fois indiquer exactement ce que vous réutilisez, et à qui/où vous l'avez pris. De plus, vous devez expliquer pourquoi vous avez eu besoin de la ressource, expliquer ce que vous avez éventuellement modifié (par exemple s'il s'agit d'un algorithme), et décrire la nature de ces modifications.

Avertissement : le non-respect de ces consignes est considéré par l'université comme du *plagiat*, et vous expose aux sanctions prévues dans son règlement.

Le système \LaTeX est couplé à BibTeX afin de gérer automatiquement les références bibliographiques du document. BibTeX facilite grandement la gestion des sources bibliographiques et leur insertion dans le rapport.

Fichier BibTeX. La procédure consiste d'abord à définir un fichier de type BibTeX décrivant chaque entrée bibliographique en détail. Le fichier `bibliographie.bib` fourni avec ce document est un exemple de fichier BibTeX. Il est placé dans le même dossier que le fichier `.tex` à compiler. Chaque entrée doit être associée à une clé unique permettant de l'identifier par la suite. Attention : n'utilisez pas de caractères accentués lorsque vous définissez vos clés.

Vous ne devez pas éditer ce fichier manuellement, car cela va causer des erreurs qui seront difficiles à détecter par la suite. Il vous faut utiliser le logiciel libre Jabref⁶, qui est prévu à cet effet. Il est écrit en Java, et est donc compatible avec les principaux systèmes d'exploitation. Il permet

6. <http://jabref.sourceforge.net/>

d'ajouter/supprimer/éditer des entrées bibliographiques, et d'éditer/générer un fichier au format BibTeX. Il permet aussi de générer les clés en préservant leur unicité.

La commande `\addbibresource`, qui est présente en début de document, permet d'indiquer l'emplacement et le nom du fichier BibTeX à utiliser. Dans le cas du document présent, on a :

```
\addbibresource{bibliographie.bib}
```

Types d'entrées. On trouve différents types d'entrées bibliographiques, dont les principaux sont les suivants :

- `Article` : article publié dans un journal [2, 4] ;
- `Book` : monographie, i.e. livre intégralement rédigé par le même groupe d'auteurs [9, 24] ;
- `InBook` : chapitre dans une monographie [8, 13] ;
- `Collection` : recueil composé de chapitres écrits par des auteurs différents, et compilés par des éditeurs [1, 12] ;
- `InCollection` : chapitre dans un recueil [3, 6] ;
- `Electronic` : page Web ou autre ressource disponible exclusivement en ligne [7, 16] ;
- `InProceedings` : article publié dans les actes d'une conférence [10, 15] ;
- `MasterThesis` et `PhDThesis` : mémoire de master ou de doctorat [5, 25] ;
- `TechReport` : travail publié en interne (à une université, entreprise, etc.) ou documentation technique [11, 14].

Les références bibliographiques (numéros entre crochets) associées à chaque type d'entrée listée ci-dessus sont des exemples, à consulter dans le fichier `bibliographie.bib`.

Champs nécessaires. Une entrée bibliographique doit toujours indiquer certaines informations, qui dépendent du type de l'entrée et prennent la forme de champs spécifiques. Ces champs sont détaillés dans la Table 6.

Dans tous les cas, il faut indiquer le DOI [19] s'il y en a un, et *sinon* l'URL pour pouvoir récupérer facilement le document concerné.

Citations. Dans le fichier `.tex`, on fait référence à une entrée bibliographique grâce à la commande `\cite` :

```
\cite{Wikibooks2010}
```

Cette commande prend en paramètre la clé unique associée à l'entrée bibliographique dans le fichier BibTeX correspondant.

Avec la mise en forme choisie pour ce modèle de document, une référence bibliographique prend la forme d'un numéro entre crochet. Pour l'exemple ci-dessus, on a le résultat suivant : [16]. Ce numéro permet de retrouver l'entrée bibliographique dans la liste située à la fin du document, qui est générée automatiquement par \LaTeX . De plus, le numéro est un hyperlien permettant d'accéder directement à cette entrée en cliquant dessus.

Il est aussi possible d'utiliser `\textcite` à la place de `\cite`, pour mentionner le nom de l'auteur. Par exemple pour `\textcite{Masuda2016}` on obtient : Masuda et Lambiotte [9]. Vous pouvez mettre plusieurs références dans une même commande `\cite`, par exemple `\cite{Wikibooks2010, Wikibooks2011}` donne [16, 17].

Attention à bien séparer la citation du texte au moyen d'un espace : c'est "une citation [2]", et non pas "une citation[2]".

Section bibliographique. BibTeX et \LaTeX vont s'occuper de numérotter les références automatiquement. La commande `\MyBibliography` permet d'insérer une liste des entrées références, généralement on la place à la fin du document (voir le code source de ce document).

Type BibTeX	Description du champ	Champ BibTeX
<i>Tous les types</i>	Nom des auteurs	author
	Titre de l'article/ouvrage	title
	Année de publication	year
	DOI	doi
	URL (seulement s'il n'y a pas de DOI)	url
@Article	Nom du journal	journal
	Numéro de volume	volume
	Numéro de sous-volume	issue
	Numéros des pages	pages
@Book/@Collection	Société publiant le livre	publisher
	Ville et pays de cette même société	address
	Titre de la série (si le livre appartient à une série)	series
	Numéro de volume dans cette série (le cas échéant)	volume
@InBook	Même champs que pour @Book, avec en plus :	
	Titre du livre	booktitle
	Numéro de chapitre dans l'ouvrage	chapter
@InCollection	Mêmes champs que pour @InBook, avec en plus :	
	Personnes ayant constitué le recueil	editor
	Numéros des pages	pages
@InProceedings	Même champs que pour @Book, avec en plus :	
	Nom de la conférence	booktitle
	Numéros des pages	pages
@Electronic	Entreprise ou institution hébergeant la ressource	organization
	URL de la ressource (obligatoire)	url
@Master/@PhDThesis	Université d'inscription	institution
	Type de mémoire (M1, M2, Doctorat, etc.)	type
@TechReport	Entreprise/université de rattachement	institution
	Type de rapport (Technique, Scientifique, etc.)	type
	Numéro de série unique du rapport	number

Table 6. Champs nécessaires aux différents types d'entrées bibliographiques BibTeX.

6 Conclusion

Il est recommandé de consulter le code source de ce document pour compléter les explications que vous venez de lire.

\LaTeX nécessite un minimum d'effort pour être pris en main, mais il n'y a rien d'insurmontable, notamment pour des étudiants en informatique (d'autant plus s'ils sont déjà familiers de langages comme HTML). Le contrôle obtenu et le rendu du document sont largement supérieurs à celui des logiciels de traitement de texte, y compris ceux commerciaux comme MS Word.

À noter qu'un très grand nombre d'extensions existent pour traiter des besoins spécifiques : diagrammes particuliers, documents de présentation de type Powerpoint, etc. Il est aussi possible d'intégrer du code Python ou R (ou autres langages) dans un document \LaTeX , qui sera réexécuté à chacune de ses compilations. À vous d'explorer toutes ces possibilités !

Références

- [1] U. BRANDES et T. ERLEBACH, éd. *Network Analysis : Methodological Foundations*. T. 3418. Lecture Notes in Computer Science. Springer, 2005. DOI : [10.1007/b106453](https://doi.org/10.1007/b106453).
- [2] J.-V. COSSU, V. LABATUT et N. DUGUÉ. « A review of features for the discrimination of Twitter users : application to the prediction of offline influence ». In : *Social Network Analysis and Mining* 6 (2016), p. 1–23. DOI : [10.1007/s13278-016-0329-x](https://doi.org/10.1007/s13278-016-0329-x).
- [3] L. DANON, J. DUCH, A. ARENAS et A. DÍAZ-GUILERA. « Community structure identification ». In : *Large Scale Structure and Dynamics of Complex Networks : From Information Technology to Finance and Natural Science*. World Scientific, 2007. Chap. 5, p. 93–113. DOI : [10.1142/9789812771681_0006](https://doi.org/10.1142/9789812771681_0006).
- [4] S. FORTUNATO. « Community detection in graphs ». In : *Physics Reports* 486.3-5 (2010), p. 75–174. DOI : [10.1016/j.physrep.2009.11.002](https://doi.org/10.1016/j.physrep.2009.11.002).
- [5] A. GERBAUD. « Modélisation de réseaux d'interaction par des graphes aléatoires ». PhD Thesis. Grenoble, FR : Université de Grenoble, 2010. URL : <http://www.theses.fr/s123781>.
- [6] V. LABATUT et J.-M. BALASQUE. « Detection and Interpretation of Communities in Complex Networks : Methods and Practical Application ». In : *Computational Social Networks : Tools, Perspectives and Applications*. Sous la dir. d'A. ABRAHAM et A.-E. HASSANIEN. Springer, 2012. Chap. 4, p. 81–113. DOI : [10.1007/978-1-4471-4048-1_4](https://doi.org/10.1007/978-1-4471-4048-1_4).
- [7] LATEX PROJECT. *LaTeX – A document preparation system*. LaTeX Project. 2010. URL : <http://www.latex-project.org/>.
- [8] K. MAINZER. « Complex Systems and the Evolution of Computability ». In : *Thinking in Complexity : The Computational Dynamics of Matter, Mind and Mankind*. 2007. Chap. 5, p. 179–226. DOI : [10.1007/978-3-540-72228-1_5](https://doi.org/10.1007/978-3-540-72228-1_5).
- [9] N. MASUDA et R. LAMBIOTTE. *A Guide to Temporal Networks*. World Scientific, 2016. DOI : [10.1142/9781786341150](https://doi.org/10.1142/9781786341150).
- [10] A. MAUTTONE, M. LABBÉ et R. FIGUEIREDO. « A Tabu Search approach to solve a network design problem with user-optimal flows ». In : *VALIO/EURO Conference on Combinatorial Optimization*. Buenos Aires, AR, 2008, p. 1–6.
- [11] D. C. PARASKEVOPOULOS, T. BEKTAS, T. Gabriel CRAINIC et C. N. POTTS. *A Cycle-Based Evolutionary Algorithm for the Fixed-Charge Capacitated Multi-Commodity Network Design Problem*. Rapp. tech. Montréal : Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, 2013, p. 1–31.
- [12] R. PASTOR-SATORRAS, M. RUBI et A. DIAZ-GUILERA, éd. *Statistical Mechanics of Complex Networks*. T. 625. Lecture Notes in Physics. Springer, 2003. DOI : [10.1007/b12331](https://doi.org/10.1007/b12331).
- [13] J. REICHARDT. « Modularity of Dense Random Graphs ». In : *Structure in Complex Networks*. T. 766. Lecture Notes in Physics. 2009. Chap. 5, p. 69–86. DOI : [10.1007/978-3-540-87833-9_5](https://doi.org/10.1007/978-3-540-87833-9_5).
- [14] M. ROSVALL et C. T. BERGSTROM. *Fast stochastic and recursive search algorithm*. Rapp. tech. Umeå, SE : Department of Physics, Umeå University, 2009.
- [15] Y.-C. WEI et C.-K. CHENG. « Towards efficient hierarchical designs by ratio cut partitioning ». In : *IEEE International Conference on Computer Aided Design*. 1989, p. 298–301. DOI : [10.1109/ICCAD.1989.76957](https://doi.org/10.1109/ICCAD.1989.76957).
- [16] WIKIBOOKS. *Guide pour la programmation en LaTeX*. Wikibooks. 2010. URL : <http://fr.wikibooks.org/wiki/LaTeX>.
- [17] WIKIBOOKS. *Guide to the LaTeX markup language*. Wikibooks. 2011. URL : <http://en.wikibooks.org/wiki/LaTeX>.
- [18] WIKIPEDIA. *Courier (typeface)*. Wikipedia. 2018. URL : [https://en.wikipedia.org/wiki/Courier_\(typeface\)](https://en.wikipedia.org/wiki/Courier_(typeface)).
- [19] WIKIPEDIA. *Digital Object Identifier*. Wikipedia. 2018. URL : https://fr.wikipedia.org/wiki/Digital_Object_Identifier.
- [20] WIKIPEDIA. *Flowchart*. Wikipedia. 2011. URL : <http://en.wikipedia.org/wiki/Flowchart>.
- [21] WIKIPEDIA. *Image matricielle*. Wikipedia. 2018. URL : https://fr.wikipedia.org/wiki/Image_matricielle.

- [22] WIKIPEDIA. *Image vectorielle*. Wikipedia. 2018. URL : https://fr.wikipedia.org/wiki/Image_vectorielle.
- [23] WIKIPEDIA. *LaTeX*. Wikipedia. 2011. URL : <http://fr.wikipedia.org/wiki/LaTeX>.
- [24] L. A. WOLSEY. *Integer programming*. New York, USA : Wiley-Interscience, 1998.
- [25] R. T. WONG. « Accelerating Benders decomposition for network design ». PhD Thesis. Massachusetts Institute of Technology, 1978.