

# Ateliers d'initiation à L<sup>A</sup>T<sub>E</sub>X- Troisième séance

Martin Van den Abbeele

Laurent Bataille

Novembre 2019

## Objectifs de la séance

- Prise en main des tableaux et des matrices
- Utilisation d'outils de mise en forme automatique
- Gestion de projets en LaTeX : comment structurer de gros documents.

## 1 Les tableaux et les matrices

Les matrices constituent la transition la plus directe entre les environnements mathématiques et les tableaux.

### Code minimaliste décrivant une matrice

```
\begin{equation*}
\mathbf{\sigma} =
\begin{matrix}
& \sigma_{11} & \sigma_{12} & \sigma_{13} \\
\sigma = & \sigma_{21} & \sigma_{22} & \sigma_{23} \\
& \sigma_{31} & \sigma_{32} & \sigma_{33}
\end{matrix}
\end{equation*}
```

Les matrices et les vecteurs s'écrivent dans les équations avec les environnements de type `matrix`, le symbole `&` est le séparateur de colonnes et `\\` sépare les lignes. Il existe différents environnements permettant de réaliser des choix esthétiques pour la représentation des matrices, qui sont chargés à partir de l'extension `amsmath`, comme `pmatrix`, `bmatrix`

Testez les variantes suivantes :

<code>\begin{matrix}</code>	<code>\begin{pmatrix}</code>	<code>\begin{bmatrix}</code>	<code>\begin{Vmatrix}</code>
<code>a &amp; b \\</code>	<code>a &amp; b \\</code>	<code>a &amp; b \\</code>	<code>a &amp; b \\</code>
<code>c &amp; d</code>	<code>c &amp; d</code>	<code>c &amp; d</code>	<code>c &amp; d</code>
<code>\end{matrix}</code>	<code>\end{pmatrix}</code>	<code>\end{bmatrix}</code>	<code>\end{Vmatrix}</code>

Les tableaux, à l'instar des images sont des flottants. Si ceci reste flou pour vous, nous vous invitons à relire les notes consacrées aux flottants à la séance précédente.

Les tableaux sont gérés nativement sous  $\text{\LaTeX}$ , cependant on se retrouve vite limité du point de vue de la mise en forme, mais les extensions suivantes permettent de corriger en partie ce détail.

### Extensions supplémentaires pour gérer les tableaux

```
\usepackage{multirow} % Fusion de cellules
\usepackage{tabularx} % Calcul automatique de la largeur des
    colonnes
\usepackage{array}      % Options de mise en forme plus é
    laborées
\usepackage{booktabs}   % Améliore le remplissage des cellules et
    permet d'affiner le style des lignes de séparation des rangées
```

Les tableaux s'insèrent en utilisant les environnements `table`, qui sert à légender et `tabular` qui contient les données du tableau en tant que telles. La commande `\label` permet de créer une référence vers le premier `\caption` précédent cette commande.

### Exemple de tableau

```
\begin{table}[h!]
\caption{Effets de l'overdose de croquettes sur des chats gourmands}
\label{tab:XP-chats}
\begin{tabular}{lcc}
\hline
\multirow{2}{*}{Sujet} & \multicolumn{2}{c}{Poids (kg)} \\
& Avant & Après \\
\hline
\hline
Mistigri & 3.58 & 4.42 \\
Garfield & 3.88 & 4.56 \\
Rourou & 3.65 & 4.69 \\
Simba & 4.12 & 4.21 \\
\hline
Tendance &  $\$3.81 \pm 0.22\$$  &  $\$4.47 \pm 0.17\$$  \\
\hline
\end{tabular}
\end{table}
```

Voici un autre exemple de tableau un peu plus simple :

TABLE 1 – Tableau très minimaliste...

1	2	3	4
5	6	7	8

- Essayez de le coder vous-même
- La position du texte dans chaque colonne est l'argument de l'environnement `tabular`. Tester l'ajout d'un séparateur vertical en ajoutant en modifiant l'argument en `l|cc`.

- Les colonnes (resp. les lignes) sont séparées par & (resp //). Pour un report de ligne à l'intérieur d'une cellule, utilisez `\newline`.
- Bordure avec `\hline`. Tester la commande `\cline{2-3}`
- Fusion de cellules via `multirow` et `multicolumn`.

En chargeant le package `multirow`, il est possible de fusionner les cellules d'un tableau. Fusionner plusieurs cellules se réalise au moyen des commandes ci-dessous :

- Fusionner des colonnes `\multicolumn{nombredecolonnes}{positionnement}{contenu}`
- Fusionner des lignes `\multirow{nombredelignes}{*}{contenu}`
- Tracer une ligne de la colonne i à la colonne j `\cline{i-j}`.

Un exemple illustrant la puissance du concept...

```
\FloatBarrier
\begin{table}[h!]
  \centering
  \caption{Fusion...}
  \subfloat[] {
    \begin{tabular}{cc|c}      % | Ligne verticale entre les colonnes
      \hline                  % Ligne horizontale
      Vegeta & Goku & & Broly \\
      \hline                  \hline
        1 & 2 & 3 & \\      % & : sépare les colonnes
        4 & 5 & 6 & \\      % \\ : sépare les lignes
      \hline
    \end{tabular}}
  \subfloat[] {
    \begin{tabular}{ccc}
      \hline
      \multicolumn{2}{c}{Gogeta} & Broly \\ \hline
      \multicolumn{2}{c}{\multirow{2}{*}{9}} & 3 \\ \hline
      & & 6 \\ \hline
    \end{tabular}
  }
\end{table}
\FloatBarrier
```

Les options (arguments) de mise en forme, `\begin{tabular}{.....}` sont résumées dans la Table 2.

TABLE 2 – Liste des options de mise en forme par colonne

l	Texte aligné à gauche	Natif
r	Texte aligné à droite	
c	Texte centré dans la colonne	
p{2cm}	Fixe la largeur de la colonne à 2cm et place le texte à gauche	
m{2cm}	Fixe la largeur de la colonne à 2cm et centre le texte verticalement	array
b{2cm}	Fixe la largeur de la colonne à 2cm et groupe le texte vers le bas	
X	Calcul automatique de la dimension de la colonne	tabularx

Vous aurez remarqué, que toutes les combinaison d’alignements horizontaux/verticaux ne sont pas couvertes par ces options... Cependant les commandes `\centering`, `raggedright` et `raggedleft` permettent de contrôler l’alignement horizontal du texte lors de l’utilisation de p, m et b. Pour injecter du texte où une commande automatiquement au début (resp. à la fin) du contenu de chaque cellule d’une colonne de type Z, il est nécessaire d’utiliser les descripteurs de textes : préciser dans les arguments du tableau `>{contenu incroyable}Z` (resp. `Z<{contenu fantastique}`). L’exemple suivant est plus parlant :

```
\begin{tabular}{|c<{\degres C}|>{l'eau }l<{.}|}|
\hline
-1 & gel \\\hline
90 & bout \\\hline
22 & est bonne \\\hline
\end{tabular}
```

Pour personnaliser le séparateur de colonnes, il est possible de remplacer les `|` par `!Texte servant de séparateur` (ou par `@Texte servant de séparateur`, testez les deux opérateurs pour comprendre la nuance) :

```
\begin{tabular}{|l!{\$ \rightarrow$}c!{\ding{37}}l|}
\hline
Robert Bidochon&RB&03.03.03.01\\\hline
Ginette Lacaille&GL&02.02.02.01\\\hline
\end{tabular}
```

Pour insérer des lignes de séparation d’épaisseur variable, il est nécessaire d’utiliser le package `booktabs`. L’exemple suivant est assez parlant :

```
\begin{table}\centering
\begin{tabular}{lrc}
\toprule[2pt]
a & b & c \\\
\cmidrule[1pt](rl){1-3}
d & e & f \\\
j & k & l \\\
\addlinespace[0.5em]
g & h & i \\\
\bottomrule[2pt]
```

```
\end{tabular}\caption{abcde}
\end{table}
```

L'utilisation de ces commandes se réalise dans la cellule, ce qui peut être fastidieux et rapidement illisible. Heureusement, l'extension `array` permet de définir de nouveaux types de colonne via la commande `\newcolumntype[nargs]{definition_du_type}...` Par exemple : `\newcolumntype{0}{>\color{blue}}`

Liste des options de mise en forme pour les séparateurs

Vous l'aurez compris, mettre en forme un grand nombre de colonnes peut se révéler rapidement fastidieux et source d'erreurs de compilation. Cependant il est possible de gérer les alignements d'un grand nombre de colonnes simultanément grâce à en utilisant la syntaxe suivante `*{nombreColonnes}{format}`

- `{|*{5}{p{2cm}}|}` est équivalent à `{|p{2cm}|p{2cm}|p{2cm}|p{2cm}\}|p{2cm}|}`
- `{|*{3}{c|}*{2}{p{4cm}}|}1*{3}{p{2cm}}|}` est équivalent à `{|c|c|c|p{4cm}|p{4cm}|}1|p{2cm}|p{2cm}|p{2cm}|}`

Vous avez désormais un aperçu global des manipulations nécessaires pour manipuler les tableaux. Les tableaux de grandes tailles restent cependant compliqués à mettre en forme manuellement. Heureusement il existe des outils pour éviter de devoir réaliser ces manipulations en lignes de code.

### Générer des tableaux $\text{\LaTeX}$ sans coder (ou presque) !

Vous l'aurez compris, écrire des tableaux en  $\text{\LaTeX}$  n'est pas l'aspect le plus fun de la rédaction quand il s'agit d'ajouter des tableaux de grandes tailles.

Heureusement d'autres personnes ont vécu cette frustration avant vous et ont créé des macros de conversion pour passer de tableaux codés en fichiers Excel/Calc en format utilisable par  $\text{\LaTeX}$ .

Ces outils sont disponibles ici :

- <https://ctan.org/tex-archive/support/excel2latex/>
- <https://sourceforge.net/projects/calc2latex/>

Le site <https://www.tablesgenerator.com> permet de réaliser la conversion en copiant collant un tableau Excel. Il est possible d'y fusionner directement des lignes et de personnaliser les séparateurs.

## 2 Les projets en $\text{\LaTeX}$

Lorsqu'un document de grande taille est créé, il convient de choisir la classe du document de façon adéquate. Typiquement le choix s'effectue entre `book`, `report` et `memoir`.

Pour rendre le code lisible et facilement manipulable, il est nécessaire de scinder le projet global en plusieurs fichiers tex. Le préambule est généralement isolé du reste du projet, tandis qu'un dossier différent est créé pour chaque partie du projet. Un set de commandes particulières permet de gérer les interactions entre les différentes parties.

Deux packages sont principalement utilisés pour réaliser ces opérations : `subfiles` ou `import`.

## 2.1 Paramétrage du projet et des sous-fichiers

Pour les gros fichiers, les codes finissent souvent par s'encombrer un peu. Il est alors possible de les partitionner Pour travailler sur plusieurs documents. Inclure un fichier dans un autre s'effectue au moyen du package **import**. Deux commandes proches existent :

### commande import

- `\import{chemin_vers_le_fichier}{nom_du_fichier.tex}`
- `\subimport{chemin_vers_le_fichier}{nom_du_fichier.tex}`

Les deux commandes fonctionnent avec des liens relatifs ou absolus. La différence réside dans le fait que la commande `\subimport` permet d'inclure un sous-fichier contenant des sous-sous-fichiers. Cela s'appelle l'inclusion imbriquée. Cette commande permet de scinder le travail mais ne divise pas la compilation.

Pour partitionner un fichier sans recompiler à chaque fois l'intégralité du projet, il est recommandé d'écrire un préambule au début de chaque fichier et d'utiliser le package **subfiles**. Ainsi il est possible compiler de petits documents.

Attention, il est important d'utiliser ce package. Il est nécessaire pour paramétrer les documents de façon à ce que l'ensemble des packages soient détectés quand on inclut des sous-documents dans un fichier maître (votre main sous overleaf),

### Utilisation de subfiles

Dans le document maître :

```
\documentclass{ma_classe_de_document}  
\usepackage{subfiles}
```

Dans les sous-fichiers :

```
\documentclass[nom_du_fichier_principal.tex]{subfiles}
```

Ceci permet de faire appel au préambule du fichier main.

## 2.2 Interactions entre les fichiers

Inclure un fichier .tex dans un autre peut s'effectuer via la commande `\subfile`.

```
\subfile{dossier/sous-dossier/mon_fichier.tex}
```

## A Comment exporter facilement des tables produites sous R vers le LaTeX ?

L'extension **stargazer** doit-être préalablement installée sous R. Cette extension permet d'exporter une grande diversité d'objets, allant du dataframe au résultat d'une régression, l'option «summary» permet de synthétiser l'information facilement.

<https://www.r-bloggers.com/stargazer-package-for-beautiful-latex-tables-from-r-statistical-models-output/>

## B Comment exporter facilement des tables produites sous Matlab vers le LaTeX ?

S'il existe qu'un outil natif pour exporter les matrices en matlab (`latex`), il n'offre pas la même ergonomie que Stargazer sous R, il n'est pas forcément très adapté aux structures et aux tables Matlab. Il existe cependant un outil mis en ligne sur le site d'échange : <https://nl.mathworks.com/matlabcentral/fileexchange/matlab-table-to-latex-conversor>

## C Changer l'orientation d'une page, passer du mode portrait en mode paysage sur une partie du document

Il est nécessaire d'insérer dans le préambule l'extension `landscape`. Tout ce qui se trouve dans l'environnement `landscape` est en paysage :

```
\begin{landscape}  
Contenu en orientation paysage  
\end{landscape}
```

## D Colorer les lignes d'un tableau

Il faut avant toute chose charger `xcolor` en plaçant à la seconde ligne.

```
\usepackage[table,x11names,dvipsnames,table]{xcolor}
```

Pour colorer une seule ligne, utiliser la commande suivante juste avant la ligne à colorer.

```
\rowcolor{couleur} % couleur : par exemple red!25, rouge avec opacité de  
25%
```

Pour faciliter la lecture d'un tableau de grande taille, il est parfois préconisé de colorer une ligne sur deux d'un tableau en insérant avant le `tabular` ou le `longtable`

```
\rowcolors{indice de début}{couleur indices pairs}{couleur indices  
impairs} % Indice débutant à 1 !
```

## E Inclure des tableaux de grande taille dans un code L<sup>A</sup>T<sub>E</sub>X

Il est nécessaire de faire appel à l'extension `longtable` et à l'environnement du même nom. `booktabs` doit également être inclus dans le document.

`longtable` ne doit pas être inclut dans un environnement `table`, on peut directement inclure la commande `\caption` à l'intérieur et pour disposer le `longtable` sur la page, il faut utiliser les environnements utilisés avec du contenu plus classique (`center`, `flushright` et `flushleft`).

Voici quelques commandes utiles :

```
\endfirsthead % fin de ligne apparaissant au sommet de la première page
\endhead % fin de la ligne apparaissant au sommet de chaque page
\endfoot % fin de la ligne apparaissant à la fin de chaque page
\endlastfoot % fin de la ligne apparaissant au pied de la dernière page
```

**Exercice :** inclure le tableau fourni dans le fichier `hugeTabColinet.tex`