LaTeX sans prise de tête

(ou presque ...)

Basile Marchand
basile.marchand@gmail.com

Juin 2013

Introduction

Ce domcument a pour vocation de fournir à tous les éléments nécessaires afin de réaliser simplement des documents LaTeX.
L'idée ici n'est pas de présenter en détail le fonctionnement et l'idéologie de LaTeX mais de fournir les éléments nécessaires à une rédaction efficace et de qualité d'un document. Certains puristes trouvront probablement certaines pratiques présentées ici douteuse comme compiler les documents avec PDFLatex mais comme il a été dit, l'objectifs n'est pas ici de faire du LaTeX dans les règles de l'art mais de faire du LaTeX efficace.
Sur ce, bonne découverte du LaTeX à tous.

LaTeX

Table des matières

Table des figures

Chapitre 1

Généralités

Résumé
Pour commencer l'apprentissage de LaTeX nous allons tout d'abord regarder un peu quelles en sont ses origines, voire l'histoire de LaTeX.
Pour ensuite, voir quels sont les éléments qu'il est nécessaire d'installer sur son ordinateur afin de réaliser des documents en LaTeX.

Contents

1.1 Histoire et intérêt

1.1.1 LaTeX par qui et pourquoi ?

LaTeX, prononcé "latek" [1], est un langage de composition de documents créé en 1983 par Leslie Lamport chercheur américain en informatique, ancien élève du MIT. Pour dire vrai, LaTeX ne constitue en réalité qu'une surcouche du langage TeX initialement développé par Donal Knut [2] en 1977.
Actuellement, LaTeX est très répandu dans le monde universitaire et scientifique, il s'agit même d'un standard internationnal pour ce qui est de la production d'articles scientifiques. Enfin, approximativement 95% des mémoires de thèses traitant de sciences "dures" sont rédigés en LaTeX, cela parce qu'en l'écriture de formules mathématiques est facilité à l'extrème et également parce qu'il est beaucoup simple de gérer des documents de taille importante en LaTeX qu'en word par exemple.

1.1.2 LaTeXvs traitement de texte classique

La différence principale entre la rédaction d'un document avec LaTeX et un traitement de texte de texte classique provient du fait qu'avec LaTeX le rédacteur n'a à se concentrer que sur la structure, le contenu de son document, la mise en page étant gérée ultérieurement lors de la phase de compilation. Alors qu'avec un traitement de texte classique (Word, Writter, ...) qui entre dans la logique du WYSIWYG (What You See Is What You Get) le rédacteur se retrouve à devoir gérer en un même temps le contenu et la forme de son document.
Ainsi nous avons dans un cas, un rédacteur qui ne doit se concentrer que sur le contenu de son document, l'organisation de ses idées alors que dans l'autre cas le même rédacteur doit à la fois gérer ses idées mais également la représentation sur le document de ses idées. Cette idée ne se retrouve pas que lors de la rédaction de document style rapport ou article. Elle est prépondérante lors de la création de présentation, comme nous le verons à la fin de ce tutoriel, il est possible d'utiliser LaTeX en remplacement de PowerPoint. L'avantage apparait alors clairement, en effet tous le monde a déjà pu assister à des présentation PowerPoint où les notions étaient noyée sous des dizaines d'animations, fondu, damier, ... En utilisant LaTeX pour créer ses présentation le rédacteur s'oblige à se focaliser sur les notions, concepts qu'il veut faire passer, les animations possibles se limitants aux apparitions et disparitions. Ainsi, on obtient finalement une présentation structurée menant à l'essentiel et faisant passer l'idée sans fioritures quelconque.

1.2 Outils nécessaires

1.2.1 Compilateur

Pour concevoir un document en LaTeX l'élément qu'il est impératif d'avoir est le compilateur, c'est celui ci qui va interpréter les fichiers `.tex` afin de générer le fichier de sortie (`.dvi`, `.ps` ou `.pdf`), c'est à dire le document rédigé et mis en page.
En d'autres mots c'est le compilateur qui va effectuer tout le travail de mise en page dont on se soustrait lorsque l'on rédige un document en LaTeX contrairement à, comme il a déjà été dit, un traitement de texte basé sur le principe du WYSIWYG où toute cette phase

1. Cette prononciation provient du fait qu'historiquement la dernière lettre est un χ grec
2. Informaticien américain et professeur émerite à l'université de Standford. Il est considéré comme l'un des pionniers de l'algorithmique. La création du langage TeX résulte d'un souci de qualité de sa part lors de la ré-édition de son ouvrage

de mise en page doit être faite par le rédacteur.
Il faut noter que le compilateur seul ne sert pas à grand chose, il est même quasiment inutile. En effet, pour rédiger un document en LaTeX il faut le compilateur ainsi que des packages, nous verons à la fin du chapitre ce que sont les packages mais pour faire simple, il s'agit de bibliothèques de fonctions. On peut voir cela comme le dictionnaire ou l'encyclopédie du compilateur, c'est à dire que l'on fourni une fonction au compilateur et lui va aller regarder dans les packages ce qu'elle signifie. Classiquement, on appelle distribution LaTeX l'ensemble compilateur-packages.
Il existe des dizaines de distributions LaTeX, elles diffèrent principalement par les packages qu'elles contiennent de base mais aussi bien entendu par le système d'exploitation pour lequel elles sont conçues, généralement les distributions LaTeX ne sont pas multi-plateforme.

1.2.2 Editeur de texte

Le second élément nécessaire à la rédaction d'un document en LaTeX est l'éditeur de texte. L'éditeur constitu "l'environnement" sous lequel va être rédigé les différents fichiers `.tex` constituant le document.
Il existe de nombreux éditeurs de texte selon le système d'exploitation sous lequel on travail, certains de ses éditeurs sont relativement basiques, c'est à dire qu'ils n'offrent comme possibilité que la coloration syntaxique. Alors que d'autres beaucoup plus évolués permettent la coloration syntaxique, l'auto-complétion ainsi que des menus d'aide syntaxique permettant d'effectuer certaines commandes LaTeX en faisant du "clic-bouton", ce que je n'encourage pas mais peut tout de même s'avérer utile pour les novices.

1.3 Installation

A présent que vous savez ce qu'il vous faut pour faire des document en LaTeX nous allons voir comment installer ces éléments sur vos ordinateurs, selon votre système d'exploitation.

1.3.1 Windows

Nous allons commencer par Windows ... Bien entendu, l'objectif ici n'étant pas de troller Windaube je me contenterai de dire que si vous pouvez travailler sous Linux ou Mac OS et bien c'est mieu !
Pour réaliser des documents LaTeX sous Windows il est vivement conseillé d'installer comme distribution, MikTeX que vous pouvez télécharger ici. Lors de l'installation faites attention à bien autoriser MikTeX à installer des packages à la volée. Sachez juste que celà vous facilitera vraiment la vie, surtout si vous n'avez pas envie de vous embetter avec la console Windows.
Ensuite pour l'éditeur de texte, le choix est plus libre. Soit vous travaillez avec TeXworks qui sera installé de base avec MikTeX, il s'agit d'un éditeur basique n'offrant comme fonctionnalité que la coloration syntaxique et l'auto-complétion dans une certaine mesure. Soit, vous pouvez télécharger Texmaker, ici, il s'agit de l'éditeur offrant le plus de fonctionnalité en ce qui concerne le LaTeX, il permet la coloration syntaxique, l'auto-complétion mais offre également un ensemble de menu afin d'assister les novices dans la rédaction d'un document en LaTeX.

1.3.2 Mac OS

1.3.3 Linux

1.4 Packages

1.4.1 Packages quésaco ?

Comme il a déjà été dit dans la partie 1.2.1, les packages constituent des bibliothèques de fonctions pour le compilateur LaTeX. En d'autres mots, de base le compilateur permet juste d'écrire un document tout simple sans aucune mise en forme particulière. Si l'on veut alors, par exemple, changer les dimensions des marges, mettre des en-têtes et pieds de pages, écrire du texte en couleur et bien deux possibilités existent :

1. Réaliser ces opérations soit mêmes en utilisant les fonctions basiques de LaTeX
2. Utiliser le travail fait par d'autre, c'est à dire appeler un package

Bien entendu ici c'est la seconde solution qui nous intéresse. Il faut savoir qu'il existe des milliers de packages, pour réaliser à peu près tout ce que l'on peu imaginer sous LaTeX. Pour utiliser un package dans un document LaTeX c'est très simple, il suffit de placer dans l'en-tête du document la commande suivante :

```
\usepackage[options]{Nom du package}
```

Comme vous pouvez le voir, il est possible d'utiliser un package en lui fournissant des options. Par exemple, pour définir les dimensions des marges soit même il faut utiliser le package geometry de la manière suivante :

```
\usepackage[left=3cm,right=3cm,top=3cm,bottom=3cm]{geometry}
```

1.4.2 A l'aide ! Il me manque un package

Chapitre 2

Premier pas avec LaTeX

Résumé
Maintenant que nous avons vu quoi installer pour faire du LaTeX, nous allons entrer dans le vif du sujet et voir comment l'on fait pour rédiger et générer un document en utilisant LaTeX.

Contents

2.1 Organisation d'un document LaTeX

2.1.1 Les classes de documents

Pour commencer, il faut savoir que lorsque l'on se met à faire du LaTeX on s'offre en réalité un outil aux possibilités multiples.
En effet, car lorsque l'on commence un document LaTeX la première chose que l'on doit renseigner, c'est la classe du document. C'est à dire que l'on doit définir quel type de document l'on souhaite rédiger. Les principales classes disponibles sont les suivantes :

- lettre : classe déstinée à la rédaction de lettre au format français
- article : classe déstinée à la rédaction d'article en vue de publications et ne contenant que quelques pages. Dans la classe article, il existe cinq niveaux de hiérarchisation qui sont :
  — section
  — sous-section
  — sous-sous-section
  — paragraphe
  — sous-paragraphe
- report : classe déstinée à la rédaction de document présentant un nombre de pages significatif tel qu'un mémoire de thèse. Dans cette classe, il existe 6 niveaux de hiérarchisation qui sont :
  — chapitre
  — section
  — sous-section
  — sous-sous-section
  — paragraphe
  — sous-paragraphe
- book : classe très semblable à la classe report si ce n'est qu'elle est déstinée à la rédaction de livres complets, plusieurs centaines de pages, elle offre certaines spécificités de mise en pages supplémentaires par rapport à la classe précédente. Dans cette classe, il existe 7 niveaux de hiérarchisation qui sont :
  — partie
  — chapitre
  — section
  — sous-section
  — sous-sous-section
  — paragraphe
  — sous-paragraphe
- beamer : classe déstinée à la réalisation de présentation, à mes yeux ayant un meilleur rendu visuel que des présentations power point. La classe beamer est extrèmement riche, elle sera abordé dans ce tutoriel de manière très superficiel. Mais elle fera l'objet, peut-être, d'un tutoriel complet.

Bien entendu, les classes qui viennent d'être présenté ne sont pas les seules existant en LaTeX il y en a de nombreuses autres et surtout il est possible de définir sa propre classe de document. Cependant l'écriture de sa propre classe représente un travail relativement important et surtout peut s'avérer légèrement hors de portée pour des "novices". De plus, étant donné les objectis de ce tutoriel, présenter la démarche d'écriture de sa propre classe ne me semblait pas pertinent.

2.1.2 Organisation d'un fichier .tex

Il faut savoir qu'un fichier `Document.tex` peut se décomposer en deux parties, l'en-tête et le corps du document.

L'en-tête

L'en-tête du fichier `Document.tex` est l'un des éléments les plus important du document car c'est elle qui va définir une grande partie de la mise en forme du document lors de la compilation. Comme nous l'avons déjà dit précédemment, c'est dans l'en-tête que l'on appel les packages que l'on souhaite utiliser pour la mise en forme. Pour rappel un package s'appel en utilisant la commande :

```
\usepackage[options]{Package}
```

Mais avant même d'appeler les packages, la première ligne de l'en-tête est celle qui défini quelle classe de document nous avons choisi. Pour cela, il faut utiliser la commande :

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

Comme pour les packages nous pouvons passer des options en argument de la commande de définition de classe. Par exemple pour réaliser ce tutoriel la première ligne de l'en-tête est la suivante :

```
\documentclass[11pt,a4paper]{report}
```

C'est à dire que nous avons utiliser un document de classe rapport donc et en options, nous avons définis une taille de police de 11 point et des pages de dimension A4.

Le corps du document

Le corps du document est très simple à repérer dans un document LaTeX car il s'agit de tous ce qui est écrit entre :

```
\begin{document}

CORPS DU DOCUMENT

\end{document}
```

Ainsi tout ce qui est écrit entre ces deux balises sera forcément présent dans le document généré après compilation.
Une astuce pour éviter de se retrouver avec un fichier `Document.tex` faisant plusieurs milliers de lignes est de décomposer ce dernier en différents fichiers `.tex`. C'est à dire que par exemple, on écrit un fichier `Chapitre_i.tex` par chapitre du document que l'on écrit et en plus dans le fichier `Document.tex` on écrit l'en-tête avec l'appel aux différents packages, etc ... Et dans le corps du document on appel les différents chapitres écrit dans les fichiers à part. Pour appeler ces fichiers externes, il suffit d'utiliser la commande `\include{Chapitre_i.tex}`. C'est à dire que nous avons :

```
\begin{document}

\include{Chapitre_1.tex}
\include{Chapitre_2.tex}
```

```
\end{document}
```

Le fait de décomposer ainsi son document permet une meilleur lisibilité lors de la rédaction et de la modification du contenu.

Attention

Il faut bien faire attention au fait que les différents fichiers `Chapitre_i.tex` ne contiennent que du texte brut. Il n'y a aucune en-tête, aucun appel de package et il n'y a pas non plus de délimiteur du document `\begin{document} ... \end{document}`.

2.2 Mise en page

A présent que nous savons comment s'organise un document LaTeX nous allons pouvoir commencer à le "remplir". Pour cela nous allons dans un premier temps nous intéressé à l'aspect visuel du document, c'est à dire tout ce qui est lié à la mise en page.

2.2.1 Dimensions des marges

Par défaut, les marges imposées dans un document LaTeX sont très (trop) grandes. Nous verrons par la suite que l'on obtient avec les marges par défaut une mise en page disgracieuse. Il est donc nécessaire d'imposer ses propres dimensions pour les marges. Pour faire cela, vous connaissez déjà la réponse car cela a été donné en exemple précédemment, il faut utiliser le package geometry en lui fournissant comme option les dimensions des différentes marges. Par exemple pour la rédaction de ce tutoriel, il a été choisi d'avoir une marge de 3 cm partout, à gauche, à droite, en haut, en bas. Cela s'est donc fait de la manière suivante :

```
\usepackage[left=3cm,right=3cm,top=3cm,bottom=3cm]{geometry}
```

2.2.2 En-tête et pied de page

Le second point concernant la mise en page du document sont les en-tête et pied de page. En effet il s'agit d'éléments importants du document que l'on rédige. Par défaut, LaTeX se contente de mettre le numéro de page au milieu du pied de page. Si l'on veut ajouter des éléments, il faut utiliser le package fancyhdr. Ensuite, pour redéfinir l'en-tête et le pied de page, il faut utiliser les commandes suivantes :

```
\usepackage{fancyhdr}                    % Appel du package
\pagestyle{fancy}                        % On défini un style de page personnalisé
\renewcommand\headrulewidth{1pt}         % On impose l'épaisseur des traits
\renewcommand\footrulewidth{1pt}         % haut et bas
%% En-tete
\fancyhead[L]{En-tête gauche}
\fancyhead[C]{En-tête centre}
\fancyhead[R]{En-tête droit}
%% Pied
\fancyfoot[L]{Pied gauche}
\fancyfoot[C]{Pied centre}
```

```
\fancyfoot[R]{Pied droit}
```

Pour mettre le numéro de page, il suffit alors de placer la commande `\thepage` là où l'on souhaite que le numéro de page soit affiché. Une autre commande intéressante est `\thechapter` elle permet de récupérer le numéro du chapitre en cours. Bien entendu cette dernière commande exist également pour les sections lorsque l'on rédige un document en classe article, il suffit alors de mettre `\thesection`.

2.2.3 Mise en forme du texte

Ce que l'on appel communément police dans la rédaction de document est en fait l'association des quattres éléments suivants :

La fonte qui correspond au dessin général des lettres

Le corps qui correspond à la taille des lettres

La forme c'est à dire romains, *italiques*, PETITES CAPITALES

La graisse médium ou gras

Choix de la fonte

Par défaut LaTeX utilise des fontes appelées *extended computer modern* créées par Donald Knuth dont nous avons déjà parlé. Dans l'idée, étant donné l'objectif visé par ce tutoriel, l'utilisation de l'ECM est largement suffisante. C'est pour cela qu'il ne sera pas expliqué ici comment changer la fonte d'un document.
Cependant, nous allons juste voir rapidement quelque exemple de changement de fontes. Tout d'abord par défaut la fonte ECM et une fonte avec empattement (ou serif) il est possible d'utiliser une fonte sans empattement (sans serif).
De la même manière, il est possible d'utiliser une fonte dite à chasse fixe, c'est à dire une fonte de type machine à écrire.

Commande	Rendu
`\textrm{...}`	Police romaine
`\textsf{...}`	Police linéale (sans sérif)
`\texttt{...}`	`Police machine à écrire`

Choix du corps

Le corps générale du document est choisi lorsque l'on déclare la classe que l'on souhaite utiliser pour le document que l'on rédige. Ensuite LaTeX gère automatiquement toutes les variations de corps pour ce qui est des titres, notes de bas de pages, etc.
Cependant, il est possible de "forcer" LaTeX à utiliser un autre corps que celui prévu, pour cela on utilise les commandes suivantes :

Commande	Rendu
`\footnotesize ... \normalsize`	Texte très petit
`\small ... \normalsize`	Texte petit
`\large ... \normalsize`	Texte grand
`\Large ... \normalsize`	Texte très grand

Choix du style

Pour finir, nous allons voir comment modifier le style, la forme du document. De base, LaTeX utilise les caractères romains. Il est cependant possible de modifier cela, pour ce faire il suffit d'utiliser les fonctions suivantes :

Commande	Rendu
`\emph{...}`	*emphase*
`\textit{...}`	*italique*
`\textsc{...}`	PETITES CAPITALES
`\textbf{...}`	**gras**

2.3 Organisation du contenu

Maintenant que nous avons vu comment modifier la police d'écriture et comment choisir ses en-tête et pied de page nous allons nous pencher sur les outils à notre dispositon pour l'organisation du contenu du document.

2.3.1 Hiérarchisation du texte

Tout d'abord, il faut noter que généralement un texte, peut importe qu'il s'agisse d'un livre, rapport, article, est strucuré par partie, sous-partie, sous-sous-partie, ... Le nombre de niveau de structuration dépendant lui en revanche de la classe du document rédigé.
Ainsi en LaTeX pour structurer le texte que l'on écrit on utilise des commandes qui vont définir le titre de la partie considéré mais également son niveau dans la hiérarchie générale du document. Ces commandes sont les suivantes :

Partie : commande `\part{Titre de la partie}`

Chapitre : commande `\chapter{Titre du chapitre}`

Section : commande `\section{Titre de la section}`

Sous-section : commande `\subsection{Titre de la sous-section}`

Sous-sous-section : commande `\subsubsection{Titre de la sous-sous-section}`

Paragraphe : commande `\paragraph{Titre du paragraphe}`

Sous-paragraphe : commande `\subparagraph{Titre du sous-paragraphe}`

Il est intéressant de noter que toutes ces commandes peuvent également être utilisées avec une étoile juste avant la première accolade, par exemple `\section*{Titre de la section}`. Cette étoile signifie juste que, pour notre exemple, la section ne sera pas numéroté. Cela peut s'avérer utile pour l'introduction d'un document par exemple.
Suivant ses "pratiques" il se peut que l'on ait envie qu'une section par exemple commence toujours en haut d'une page. Pour cela, il suffit de forcer le passage à la page suivante. Cela se fait avec la commande `\newpage`. Il est également possible d'être plus subtil, en faisant une suggestion de saut de page ou non à LaTeX qui ensuite gèrera s'il convient ou non de suivre cette suggestion. Ainsi pour suggérer à LaTeX d'aller sur une nouvelle page il faut taper la commande `\pagebreak` et au contraire pour lui suggérer de rester sur la même page il faut utiliser la commande `\nopagebreak`.

2.3.2 Mise en forme "avancée"

Après avoir vu comment nous povions hiérarchiser notre document, nous allons revenir quelque peu sur la mise en forme du document. En effet, il peut arriver qu'à certains endroit du document l'on veuille attirer l'attention sur un point qui nous semble particulièrement important, critique. La solution la plus évidente pour y parvenir est alors de mettre un peu de couleur. Il faut savoir que LaTeX a pour vocation l'écriture d'ouvrage scientifique, c'est à dire des documents sobres. C'est pour cela que l'utilisation des couleurs dans un document LaTeX est très simple uniquement lorsque l'in veut faire une "petite touche" de couleur. Par exemple, il va être plutôt compliqué de mettre en place des titres de couleurs avec en plus une couleur différentes par niveau de titre. C'est faisable mais compliqué. De la même manière, réaliser un tableau en LaTeX avec des cases ayant un fond coloré, comme on peut très souvent le voir dans les document word, n'est pas conseillé.
C'est pour cela, que nous n'allons ici que nous intéresser à la méthode pour mettre un morceau de texte en couleur, en utilisant une couleur prédéfinie ou en définissant sa couleur. La commande pour mettre du texte en couleur est la suivante :

```
\textcolor{couleur}{texte à colorier}
```

Par contre, comme il a été dit précédemment LaTeX n'a pas été prévu pour faire des documents folkloriques, c'est pour cela que de base seul les trois couleurs primaires, RGB, sont connues par LaTeX. Pour utiliser d'autres couleurs nous pouvons, soit la définir, pour cela on place dans l'en-tête, la commande.

```
\definecolor{Nom de la couleur}{rgb}{indice de couleur rgb}
```

Pour déterminer les indices de couleur rgb il vous suffit d'aller sur internet et vous pouvez trouver des palettes avec des centaines de couleurs, vous trouverez forcément celle qu'il vous faut.
La seconde solution, consiste à utiliser la package xcolor. Pour rappel, l'appel à ce package se fait en placant dans l'en-tête du fichier principale la commande :

```
\usepackage{xcolor}
```

L'avantage de ce package est que de nombreuses couleurs usuelles ont déjà été définies ce qui vous dispense de rechercher les indices de couleur rgb. Pour connaitre toutes les couleurs définies dans le package xcolor, je vous invite à aller regarder la documentation du package.

En typologie française, un texte se décompose en plusieurs paragraphes, chaque paragraphes commencant par un alinéa à gauche, ou indentation et se termine par un alinéa à droite. Bien entendu LaTeX respecte cette règle typologique, ainsi pour préciser que l'on change de paragraphe, il faut laisser deux lignes ou plus de vides dans le fichier sources. Si on le souhaite, on peut forcer LaTeX à ne pas pas mettre d'alinéa, pour cela il suffit de mettre devant la première ligne du paragraphe en question la commande `\noindent`. Cela donne ce que l'on appel une composition en pavé. Nous pouvons également forcer le passage à la ligne dans un paragraphe en utilisant soit la commande `\newline`, soit la commande `\\`.

2.3.3 Les listes

Un outils très utile dans la réaction de document est la liste numérotée, ou non. En effet, il est toujours utile de procéder à une énumération sous forme de liste cela permet de cibler les éléments principaux. Avec LaTeX il est possible de réaliser deux types de listes :

1. Les listes numérotées
— Les listes non-numérotées

Listes non-numérotées

Il existe deux styles de liste non-numérotées, les listes à puces et les listes que l'on va qualifier de description.
Pour réaliser une liste à puce, il faut procéder de la manière suivante :

```
\begin{itemize}
\item Elément numéro 1
\item Elément numéro 2
\item Elément numéro 3
\end{itemize}
```

— Elément numéro 1
— Elément numéro 2
— Elément numéro 3

Bien entendu il est possible de remplacer les tirets de l'environnement itemize par un autre style de puce. Pour cela il faut placer dans l'en-tête du document la commande suivante :

```
\renewcommand{\labelitemi}{nouvelle puce}
```

Si l'on fait attention, on peut remarquer que l'on a écrit | le i à la fin provient du fait que les listes qu'elles soient numérotées ou non peuvent être imbriquées les unes dans les autres. On peut ainsi définir des listes à différents niveaux et on peut dans le cas d'une liste de type itemize, définir une puce différentes pour chaque niveaux de la liste. Par exemple si l'on ajoute dans l'en-tête du document les commandes suivantes :

```
\renewcommand{\labelitemi}{$\bullet$}
\renewcommand{\labelitemii}{$\circ$}
\renewcommand{\labelitemiii}{$\diamonds$}
\renewcommand{\labelitemiv}{$\ast$}
```

- Niveau 1
  - Niveau 2
    - Niveau 3
      - Niveau 4

Ensuite pour réaliser une liste descriptive, il faut procéder de la manière suivante :

```
\begin{description}
\item[Elément 1] est le premier élément
\item[Elément 2] est le second élément
\item[Elément 3] est le troisième élément
\end{description}
```

Elément 1 est le premier élément
Elément 2 est le second élément
Elément 3 est le troisième élément

Listes numérotées

En complément des listes que l'on vient de voir, il existe également les listes numérotés. Pour réaliser une liste numérotée, il faut procéder de la manière suivante :

```
\begin{enumerate}
\item Elément numéro 1
\item Elément numéro 2
\item Elément numéro 3
\end{enumerate}
```

1. Elément numéro 1
2. Elément numéro 2
3. Elément numéro 3

Si on le souhaite il est possible de changer l'apparence de la numérotation, on peut par exemple mettre des chiffres romains ou bien des lettres minuscules. Pour faire cela, il est nécessaire d'utiliser le package enumerate.

```
\usepackage{enumerate}
```

Il suffit alors d'utiliser l'environnement enumerate de la manière suivante :

```
\begin{enumerate}[I]
\item Elément numéro 1
\item Elément numéro 2
\item Elément numéro 3
\end{enumerate}
```

I Elément numéro 1
II Elément numéro 2
III Elément numéro 3

Ou bien encore :

```
\begin{enumerate}[(a)]
\item Elément numéro 1
\item Elément numéro 2
\item Elément numéro 3
\end{enumerate}
```

(a) Elément numéro 1
(b) Elément numéro 2
(c) Elément numéro 3

Comme pour les listes non-numérotées, il est tout a fait possible d'imbriquer une liste dans une autre, en revanche ici le changement du style de numérotation se fait automatiquement. Cela peut par exemple donner :

```
\begin{enumerate}
\item Niveau 1
\begin{enumerate}
\item Niveau 2
\begin{enumerate}
\item Niveau 3
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

1. Niveau 1
   (a) Niveau 2
      i. Niveau 3

Pour finir, il est tout a fait possible de mélanger listes numérotées et non-numérotées sur plusieurs niveaux. Par exemple, on peut avoir :

```
\begin{enumerate}
\item Elément 1
\item Elément 2
\begin{itemize}
\item Elément 2.1
\begin{enumerate}
\item Eléments 2.1.(i)
\item Eléments 2.1.(ii)
\end{enumerate}
\item Elément 2.2
\end{itemize}
\end{enumerate}
```

1. Elément 1
2. Elément 2
   — Elément 2.1
      (a) Eléments 2.1.(a)
      (b) Eléments 2.1.(b)
   — Elément 2.2
3. Elément 3

Pour information
En typographie française, chaque ligne d'une liste numérotée commence par une lettre majuscule et chaque ligne d'une liste non numérotée commence par une lettre minuscule.

2.4 Un premier document tout simple

Chapitre 3

Rédaction d'un document scientifique en LaTeX

Résumé

A présent que nous avons vu comment rédiger un document basique sous LaTeX tout en pouant tout de même choisir de nombreux paramètres, nous allons nous pencher sur les outils nécessaires à la rédaction de documents scientifiques en LaTeX. Bien entendu tout ce qui va être présenté ici n'est pas uniquement dédié à une utilisation ans un contexte scientifique, il est tout a fait possible si le besoin s'en fait ressentir de transposer ce qui va être vu ici à d'autres domaines.

Contents

3.1 Les flottants

3.1.1 Introduction

A présent que nous savons comment rédiger un document simple en LaTeX nous allons nous pencher sur les outils nécessaires à la rédaction d'un document scientifique. Tout d'abord si l'on réfléchi un peu, nous allons rapidement constater qu'un document scientifique n'est jamais constitué que de texte, il y a très souvent des images et des tableaux. Il s'agit donc de ce que nous allons voir de suite, c'est à dire comment insérer une image ou un tableau dans un document LaTeX.

Et donc pour réaliser cela, il a été introduit dans LaTeX le concept de flottant. L'idée de base est que les éléments tels que les images et les tableaux ne rentrent pas sur une ligne. Il faut donc réussir à gérer le placement de tels entitées au sein du document tout en conservant une mise en page agréable. Si l'on prend l'exemple d'une image que l'on veut insérer en bas d'une page mais cette dernière est trop grande et ne rentre pas. Avec un logiciel tel que Word l'image va "pousser" le texte qui est après elle, jusqu'à ce qu'elle arrive en haut de la page suivante, laissant ainsi un blanc important sur la page initialement prévu pour la figure. Alors qu'avec LaTeX, l'image va certe passer en haut de la page suivante, cependant elle ne vas pas décaler le texte. C'est à dire qu'une partie du texte qui suivait initialement l'image va se retrouver avant cette dernière. Cela peut déranger au début surtout lorsqu'il ya beaucoup d'image mais l'on s'y habitue rapidement et de plus avec le système de label et référence de LaTeX, que l'on verra dans la suite de ce chapitre, il est très facile de faire des renvoie vers les ifférentes figures au sein du corps du document.

3.1.2 Les figures

Le premier flottant que nous allons voir et l'envirronement figure. La commande complète pour insérer une figure est la suivante :

```
\begin{figure}[Consigne de placement]
\centering\includegraphics[dimensions]{chemin de l'image}
\caption{Titre de la figure}
\end{figure}
```

Comme il a été dit en introduction, les flottant ont été concu à la base pour gérer le placement des images automatiquement. Il parait alors étrange de devoir fournir une consigne de placement. En réalité, il serait plus correct de parler de recommandations de placement plutôt que de consigne. C'est à dire qu'à travers la consigne de placement, on va faire une proposition à LaTeX qui va ou non en tenir compte lors de la génération du document. Les consignes de placement possibles sont les suivantes :

h signifie que l'on veut placer le flottant ici, c'est à dire à l'endroit où il apparait dans le fichier `.tex`.

t signifie que l'on veut placer le flottant en haut de la page.

b signifie que l'on veut placer le flottant en bas de la page.

p signifie que l'on veut placer le flottant sur une page à part.

Ensuite les dimensions de l'image peuvent être imposées de plusieurs manières différentes. Tout d'abord, il est possible d'imposer une échelle, pour cela il suffit d'écrire `\includegraphics[scale=0.5]` ce qui par exemple insérera dans le document une image à l'échelle 1/2. L'inconvénient de cette méthode est qu'elle donne finalement un document avec des images qui ont toutes des tailles différentes ce qui n'est pas vraiment harmonieux.

Attention
Il est important de remarquer que pour imposer l'échelle de l'image, il a été utilisé les notations anglo-saxonnes et non pas les notations françaises. En d'autres mots, dans LaTeX le séparateur des décimales n'est pas une virgule mais un point.

Le plus simple est donc d'imposer directement une des deux dimensions de l'image, soit la largeur, soit la hauteur, LaTeX s'arrange ensuite avec la dimension qui est laissée libre afin de ne pas déformer l'image. Pour faire cela, il suffit de tapper la commande `\includegraphics[width=8cm]{chemin de l'image}` pour avoir une image de 8 cm de largeur ou bien `\includegraphics[height=8cm]{chemin de l'image}` pour avoir une image de 8 cm de hauteur.

Pour information
Précédemment nous venons d'imposer une dimension d'image en utilisant des centimètres. Il est possible de faire cela avec toutes les unités du système métrique. Mais il est également envisageable de définir la largeur de l'image par exemple, comme étant un pourcentage de la largeur du texte. Pour cela il suffit d'écrire la commande `\includegraphics[width=0.4\textwidth]{chemin de l'image}`, ce qui donne une image ayant pour largeur 40% de la largeur du texte.

Pour finir, il faut faire attention au chemin de l'image, il s'agit souvent d'une source d'erreur. Pour qu'il n'y ait pas de problème, il faut s'assurer que les noms dans le chemin ne contiennent ni espace, ni caractères spéciaux tels que les accents par exemple. Si l'image que l'on souhaite insérer se trouve dans le même dossier que le fichier principale `Document.tex` ou dans un dossier enfant, il suffit juste de mettre le chemin depuis `Document.tex` jusqu'à l'image en question. Alors que si l'image que l'on veut insérer est totalement ailleur, il est nécessaire de repartir de la racine du système, par exemple avec Windows, `C:\...`
Pour plus de clartée dans la gestion des images, je conseil de créer à côté du fichier principale `Document.tex` un dossier que l'on appel `Images` par exemple. Ensuite dans ce dossier on créé un sous-dossier par fichier `Chapitre_i.tex` et l'on nomme ce sous-dossier du même nom que le fichier `.tex` auquel il est "lié". Par exemple, je vais ici insérer l'image `cerise.png`. Elle est placé dans le dossier `Chapitre_3` lui-même contenu dans le dossier `Images`. La commande à tapper est alors la suivante :

```
\begin{figure}[h]
\centering\includegraphics[width=0.5\textwidth]{Images/Chapitre_3/cerise}
\caption{Exemple d'insertion d'image avec des cerises}
\end{figure}
```

Cela donne le résultat que l'on peut voir à la figure 3.1. Vous l'aurez peut être remarqué mais il n'a pas été nécessaire de donner l'extension de l'image. En effet, LaTeX trouve automatiquement la bonne image, ainsi cela permet d'avoir la même image avec le même nom mais des formats différents dans le même dossier sans que LaTeX ne soit "perturbé". Au niveau du format des images, cela dépend de la manière dont on compile les fichiers `.tex`.

- si l'on compile avec LaTeX est bien les seules images acceptables sont des images au format Post Script (`.ps` ou `.eps`).
- alors que si l'on compile avec pdfLaTeX est bien les images acceptables sont des images au format, `.png`, `.jpg` ou `.jpeg` et `.pdf`.

Pour finir, dans certains cas, on peut vouloir mettre des images côte à côte le tout sous une seule et même légende. Pour cela il faut utiliser le package subfig, il suffit de placer

FIGURE 3.1 – Exemple d'insertion d'image avec des cerises

dans l'en-tête du fichier principal la ligne :

```
\usepackage{subfig}
```

Ensuite dans le corps du document pour mettre différentes figures sous la même légende, il faut procéder de la manière suivante :

```
\begin{figure}[h]
\centering
\subfloat[Cerises]{\includegraphics[width=0.33\textwidth]{Images/Chapitre_3/cerise}}
\subfloat[Fraises]{\includegraphics[width=0.33\textwidth]{Images/Chapitre_3/fraise}}
\subfloat[Melon]{\includegraphics[width=0.33\textwidth]{Images/Chapitre_3/melon}}
\caption{Différents fruits d'été}
\end{figure}
```

(a) Cerises (b) Fraises (c) Melon

FIGURE 3.2 – Différents fruits d'été

Pour finir, nous allons voir comment faire pour que le corps du texte "coule" autour d'une image. En effet, dans certains cas il se peut que placer une image avec seulement du texte avant et après soit peu esthétique, par exemple une image très élancé va conduire à une page qui sera presque vide en plein milieu du document. Il s'avère alors intéressant de pouvoir placer du texte à côté de l'image. Pour faire cela il faut utiliser le flottant wrapfigure, une variante du flottant figure. La syntaxe à utiliser est la suivante :

```
\begin{wrapfigure}[NLigne]{placement}{dimensions}
\centering\includegraphics[dimensions]{chemin de l'image}
\caption{Légende}
\end{wrapfigure}
```

Les différents paramètres de l'environnement wrapfigure ont la signification suivante :

NLigne : nombre de ligne qui devront se retrouver à coté de la figure.

placement : précise si la figure devra être sur la gauche (l), sur la droite (r) ou au centre (c) de la page.

dimensions : désigne la largeur de la zone ne devant pas être occupé par du texte. Cette dimension est nécessairement supérieure ou égale à la largeur de l'image que l'on insère.

Si l'on considère un exemple :

```
\begin{wrapfigure}[15]{r}{0.5\textwidth}
\centering\includegraphics[0.4/textwidth]{Images/Chapitre_3/cerises}
\caption{Exemple de wrapfigure avec des cerises.}
\end{wrapfigure}
Du texte qui se répète encore et encore ... Du texte qui se répète encore
et encore ... Du texte qui se répète encore et encore ... Du texte qui se
répète encore et encore ... Du texte qui se répète encore et encore ... Du
texte qui se répète encore et encore ... Du texte qui se répète encore et
encore ... Du texte qui se répète encore et encore ... Du texte qui se répète
encore et encore ... Du texte qui se répète encore et encore ... Du texte qui
se répète encore et encore ... Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
```

Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...
Du texte qui se répète encore et encore ...

FIGURE 3.3 – Exemple de wrapfigure avec des cerises.

Du texte qui se répète encore et encore ... Du texte qui se répète encore et encore ... Du texte qui se répète encore et encore ... Du texte qui se répète encore et encore ...

Attention

Le texte que l'on veut placer à côté de la figure doit impérativement être placé après l'environnement wrapfigure et collé à ce dernier.
Le fait de mettre une image et du texte côte à côte fait partie des quelques petites choses que le compilateur LaTeX n'apprécie pas trop. Pour cela la prise en main et l'utilisation de lenvironnement wrapfigure peut demander un certain temps ...

3.1.3 Les tableaux

A présent que nous avons vu l'essentiel au sujet des figures, nous allons nous intéresser au second type de flottant prédéfini dans LaTeX, à savoir les tableaux.

Le principe est le même que pour les figures, pour définir un flottant tableau il suffit de procéder de la manière suivante :

```
\begin{table}[h]
\begin{center}
un beau tableau
\end{center}
\caption{Environnement flottant pour tableau}
\end{table}
```

un beau tableau

TABLE 3.1 – Environnement flottant pour tableau

Il ne reste plus qu'à définir le tableau dans cet environnement. Pour ce faire, il existe plusieurs méthodes ais nous allons dans un premier temps nous concentrer sur celle implémentée de base dans LaTeX. La réalisation de tableaux basique se fait par l'intermédiaire de l'environnement tabular.

```
\begin{table}[h]
\begin{center}
\begin{tabular}{colonnes}
lignes
\end{tabular}
\end{center}
\caption{Environnement flottant pour tableau}
\end{table}
```

Le paramètre colonnes est égal à n lettres choisi parmi (l,r,c,p) où n correspond au nombre de colonnes du tableau que l'on veut écrire. Les différentes lettres correspondent à :

l : le contenu de la colonne est aligné à gauche

r : le contenu de la colonne est aligné à droite

c : le contenu de la colonne est centré

p{largeur} : la largeur de la colonne est fixé, le contenu sera écrit sur plusieurs lignes. le paramètre largeur est une distance donné classiquement en centimètres.

Et ensuite à la place de ligne, on renseigne les différentes valeurs du tableau ligne par ligne en séparant les différentes colonnes par des éperluette & et en signifiant un retour à la ligne par un double backslash.

Par exemple, la description suivante :

```
\begin{table}[h]
\begin{center}
\begin{tabular}{lll}
   1 & 2 & 3 \\
   4 & 5 & 6 \\
\end{tabular}
\end{center}
\caption{Un premier exemple de tableau}
\end{table}
```

Donne le résultat suivant : On remarque alors tout de suite que le tableau ainsi obtenu

1	2	3
4	5	6

Table 3.2 – Un premier exemple de tableau

n'est pas pratique car il n'y a aucune délimitation des lignes et des colonnes. Pour réaliser cela c'est très simple, la séparation des colonnes se fait en ajoutant des pipes (barres verticales) entre les différents paramètres des colonnes et la séparation entre les lignes se fait en ajouant la commande `\hline` entre les lignes. Par exemple :

```
\begin{table}[h]
\begin{center}
\begin{tabular}{|l|l|l|}
\hline
1 & 2 & 3 \\
\hline
4 & 5 & 6 \\
\hline
\end{tabular}
\end{center}
\caption{Un second exemple de tableau}
\end{table}
```

1	2	3
4	5	6

Table 3.3 – Un second exemple de tableau

Bien entendu, il est tout à fait possible de se faire suivre deux pipes ou deux `\hline` afin d'avoir des lignes doublées.

Enfin, il arrive fréquemment que l'on souhaite fusionner des lignes ou des colonnes d'un tableau. Pour cela on procède de la manière suivante :

Fusionner des colonnes

On utilise la commande `\multicolumn{nombre}{colonnes}{contenu}` où nombre représente le nombre de colonne que l'on souhaite fusionner, colonnes représente comme pour tabular l'alignement du contenu de la colonne et texte représente le contenu dela colonne. Si l'on considère l'exemple suivant :

```
\begin{table}[h]
\begin{center}
\begin{tabular}{|l|c|c|}
   \hline
   colonne 1 & \multicolumn{2}{c|}{colonnes 2 \& 3} \\
   \hline
  1 & 2 & 3 \\
   \hline
   4 & 5 & 6 \\
   \hline
\end{tabular}
\end{center}
\caption{Un exemple de tableau avec fusion de deux colonnes}
\end{table}
```

colonne 1	colonnes 2 & 3	
1	2	3
4	5	6

Table 3.4 – Un exemple de tableau avc fusion de deux colonnes

Pour finir, il est possible de réaliser une fusion de plusieurs lignes, pour cela on procède de la manière suivante :

```
\begin{table}[h]
\begin{center}
\begin{tabular}{|l|c|r|}
   \hline
    colonne 1 & colonne 2 & colonne 3 \\
    \hline
    1 & 2 & 3 \\
    \cline{2-3}
       & 5 & 6 \\
    \hline
\end{tabular}
\end{center}
\caption{Un exemple de tableau avc fusion de deux lignes}
\end{table}
```

3.1.4 Créer ses flottants

Si l'on résume, nous venons de voir qu'il existe dans LaTeX ce que l'on appel des flottant servant à placer les figures et les tableaux. Si l'on simplifie les choses, il est possible de dire

colonne 1	colonne 2	colonne 3
1	2	3
	5	6

Table 3.5 – Un exemple de tableau avc fusion de deux lignes

que les flottant sont des "boites" dans lesquels on mets des figures ou des tableaux qui permettent de donner un nom, une légende, à ces éléments et de les numéroter. L'avantage que cela représente, nous le verons par la suite, et que l'on va ainsi pouvoir générer automatiquement des tables des figures et des tableaux dans notre document.

La question que l'on peut alors se poser est, n'y a t il pas également d'autres éléments qui pourrait nécessité l'utilisation d'un flottant ? Et si oui, comment pouvons nous faire ?

La réponse à la première question est oui, bien sur qu'il existe d'autre contenu pouvant nécessiter un flottant. Par exemple dans la suite de ce chapitre noue allons voir comment tracer des courbes en LaTeX, il parait lors plus logique d'avoir un flottant courbe nous permettant ainsi de faire une numérotation particulière des courbes tracées et un index de ces courbes plutot que de les placer dans un flottant figure et ainsi les mélanger au reste des images. De la même manière, nous verrons au chapitre suivant comment représenter des organigrammes ainsi que du pseudo-code [1] choses utiles lorsque l'on touche au développement informatique. Et donc il apparait là aussi comme utile de définir des flottant particuliers pour ces éléments.

Bien entendu, nous avons dans LaTeX tous les outils nécessaires à la mise en place de nos propres environnement flottant. Pour cela, la première chose à faire est d'appeler le package float. Pour rappel, cela se fait en placant la commande `\usepackage{float}` dans l'en-tête du document. Ensuite, la définition d'un nouvel environement flottant se fait à l'aide des trois commande suivante :

```
\floatstyle{style}
\newfloat{Nom}{Positionnement}{extension}[numérotation]
\floatname{Nom}{Désignation}
```

La signification des différents paramètres est la suivante :

style permet de définir le style graphique du flottant. Les valeurs possibles sont plain, plaintop, boxed et ruled.

Nom défini le nom de l'environnement flottant, le nom ainsi choisi sera le terme à mettre dans `\begin{Nom}[htpb] ... \end{Nom}` pour insérer un flottant

Positionnement représente les paramètres de positionnement, h, t, b, p qu'il sera possible d'utiliser avec le nouveau flottant

extension , il s'agit de l'extension qui sera utilisé lors de la création du fichier permettant de recenser tous les flottants, leurs numérotation et leurs légende. Et ce pour chaque type de flottant. Par exemple il existe de base des fichiers `.lof` pour List Of Figure.

numérotation défini si la numérotation du nouvel environnement flottant doit dépendre d'un autre compteur, par exemple comme pour les figures, la numérotation inclue le numéro du chapitre où se situe la figure.

Désignation permet de définir le nom qui apparaitra avec la numérotation en légende du flottant

1. Algorithme de calcul ayant un formalisme et une syntaxe proche de ce que l'on trouve dans les languages informatiques de bas niveau tels que le C.

Si par exemple on place dans l'en-tête du document les lignes suivantes :

```
\floatstyle{plain}
\newfloat{ExempleA}{htbp}{exA}[chapter]
\floatname{ExempleA}{Exemple A}
```

Et qu'ensuite on écrit :

```
\begin{ExempleA}[h]
Un contenu quelconque
\caption{Un exemple de flottant avec un style plain}
\end{ExempleA}
```

Un contenu quelconque

Exemple A 3.1 – Un exemple de flottant avec un style plain

Ensuite, un autre exemple, si l'on écrit les lignes suivantes dans l'en-tête du document

```
\floatstyle{ruled}
\newfloat{ExempleB}{htbp}{exB}[section]
\floatname{ExempleB}{Exemple B}
```

Et qu'ensuite on écrit :

```
\begin{ExempleB}[h]
Un contenu quelconque
\caption{Un exemple de flottant avec un style ruled et une numérotation
basée sur celle des sections}
\end{ExempleB}
```

Exemple B 3.1.1 Un exemple de flottant avec un style ruled et une numérotation basée sur celle des sections

Un contenu quelconque

3.2 Les équations

3.2.1 Introduction

A présent que nous avons vu comment insérer des images à l'aide de flottant, nous allons passer à l'autre point important dans la réalisation d'un document scientifique, la rédaction de formules mathématiques.

Il faut noter que la rédaction de formules mathématique est le point fort de LaTeX c'est vraiment pour cela qu'il est extrèmement répandu dans le monde acédémique et de la recherche. Les possibilitées offertes par LaTeX pour la rédaction d'équation étant quasi-illimitées ils va s'avérer être difficile de balayer ici toutes les fonctions existant. Cependant, nous allons essayer de présenter les commandes les plus courantes ainsi que quelques commandes moins classiques mais permettant une certaine amélioration des équations classiques

Pour les novices en LaTeX, tout l'intéret d'utiliser TeXmaker apparait lors de la rédaction d'équation. Car comme il a déjà été dit, on trouve dans TeXmaker un menu permettant

d'obtenir la commande associé à quasiment tous les symboles mathématiques. Il n'y a qu'à cliquer sur le symbole pour ajouter la commande correspondante dans le fichier `.tex`.

Vous vous en doutez peut être, maintenant que vou s commencer à connaitre LaTeX, mais pour écrire une équation il faut la placer dans un environnement adéquat. Attention, il ne s'agit pas d'un flottant. Cependant cet environnement va avoir pour fonction de définir la mise en page de l'équation. Dans la pratique, on distingue deux mises en pages différentes pour les équations :

En ligne , c'est à dire que l'équation va être placé dans une ligne de texte. Par exemple, la solution de l'équation $x - 3 = 0$ est $x = 3$.

Hors ligne , c'est à dire que là l'équation va être mise sur une ligne à part et centrée sur cette ligne. Par exemple, la solution de l'équation

$$x - 3 = 0$$

est

$$x = 3$$

Pour définir une équation en ligne, il faut encadrer l'expression de cette dernière par le sigle dollar $. Si l'on reprend l'exemple précédent, cela donne :

```
la solution de l'équation $x-3=0$
est $x=3$.
```

la solution de l'équation $x-3 = 0$ est $x = 3$.

En revanche, pour définir une équation hors-ligne il existe plusieurs solutions. En fait tout dépend de si l'on souhaite ou non que l'équation soit numérotée. L'intéret d'avoir une équation numéroté est que, nous verrons cela en détails à la fin de ce chapitre, nous allons pouvoir faire des référence et des renvois vers ces équations, facilitant ainsi grandement la compréhension du lecteur.

Pour définir une équation numéroté, on procède de la manière suivante :

```
\begin{equation}
x-3=0
\end{equation}
```

$$x - 3 = 0 \tag{3.1}$$

La solution pour supprimer la numérotation de cette équation est très simple, il s'agit en fait de la même solution que pour supprimer la numérotation d'un chapitre, d'une section, ... Il suffit d'ajouter une étoile $*$ dans le nom de l'environnment. Cela donne en pratique :

```
\begin{equation*}
x-3=0
\end{equation*}
```

$$x - 3 = 0$$

Pour définir une équation sans numérotation, il existe une méthode plus simple qui consiste à encadrer l'équation par `\[...\]`. Dans la pratique cela donne :

```
\[ x-3=0 \]
```

$$x - 3 = 0$$

Et bien maintenant que vous savez comment dire au compilateur LaTeX que vous avez écrit une équation, nous allons voir commen écrire concrètement cette équation.

3.2.2 Symboles

Pour commencer nous allons ici fournir une liste non exhaustive de symboles mathématiques utilisable avec LaTeX.

Lettres grecs

Les premiers symboles que nous allon voir sont les lettres de l'alphabet grec. En effet, il est très courant d'utiliser cet alphabet lors de l'écriture d'équations mathématiques.

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

Table 3.6 – Alphabet grec

Symboles de flèches

D'autres symboles qu'il est relativement utile de connaitres sont les flèches. En effet, elles sont très utiles pour clarifier une démarche de réflexion, résolution.

\leftarrow	$\leftarrow$	\uparrow	$\uparrow$	\hookleftarrow	$\hookleftarrow$	\nearrow	$\nearrow$
\Leftarrow	$\Leftarrow$	\Uparrow	$\Uparrow$	\hookrightarrow	$\hookrightarrow$	\searrow	$\searrow$
\rightarrow	$\rightarrow$	\downarrow	$\downarrow$	\leftharpoonup	$\leftharpoonup$	\swarrow	$\swarrow$
\Rightarrow	$\Rightarrow$	\Downarrow	$\Downarrow$	\rightharpoonup	$\rightharpoonup$	\nwarrow	$\nwarrow$
\leftrightarrow	$\leftrightarrow$	\updownarrow	$\updownarrow$	\leftharpoondown	$\leftharpoondown$		
\Leftrightarrow	$\Leftrightarrow$	\Updownarrow	$\Updownarrow$	\rightharpoondown	$\rightharpoondown$		
\mapsto	$\mapsto$	\leadsto	$\leadsto$	\rightleftharpoons	$\rightleftharpoons$		

Table 3.7 – Flèches

La particularité des éléments de la première colonne, est qu'il est possible d'en avoir des versions "allongées". Pour cela, il suffit de modifier le mot clé en lui ajoutant le préfixe `\long` ou `\Long` suivant si le mot clé initial commence, respectivement, par une voyelle ou une consonne. Il faut faire attention au fait que par exemple il faut écrire `\Longrightarrow` et non pas `\LongRightarrow`. D'ailleur cela illustre une règle générale, il n'y a jamais de lettre majuscule au milieu d'une commande LaTeX.

Délimiteurs

Ensuite, il existe plusieurs types de délimiteurs en LaTeX et l'utilisation de ces derniers est assortie de certaines règles qu'il est écessaire de respecter.

Ponctuation et polices

3.2.3 Equations scalaires et systèmes d'équations

Les premiers éléments que nous allons voir, sont la rédaction d'équation scalaire ainsi que l'écriture de systèmes d'équations.

Comme vous avez pu le voir précédemment, l'écriture d'une équation simple, comme l'équation 3.1 ne fait pas intervenir de commande particulière, la saisie d'une telle équation est très instinctive. Vous avez pu remarquer que dans l'équation 3.1 il n'a pas été représenté le symbole multiplier. Si l'on souhaite représenter tous les symboles, les commandes sont les suivantes :

	addition	soustraction	multiplication	division
symbole	`$a+b$`	`$a-b$`	`$a\times b$`	`$\frac{a}{b}$`
résultat	$a+b$	$a-b$	$a \times b$	$\frac{a}{b}$

Vous avez ainsi à votre disposition les quatre symboles usuels pour l'écriture d'équation scalaire en LaTeX. Mis à part certains opérateur que nous allons voir dans a partie qui suit, vous n'aurez pas besoin de beaucoup plus d'éléments pour la rédaction d'équation. La principale "difficulté" provient de la mise en forme que l'on va souhaiter pour ces équations.

Par exemple considérons que l'on veuille montrer que $(a+b)(a-b) = (a^2 - b^2)$ en détaillant toutes les étapes du calcul. La première solution que l'on peut envisager est la suivante :

```
\begin{equation}
(a+b)(a-b) = a^2 -ab + ab - b^2
\end{equation}
\begin{equation}
(a+b)(a-b) = a^2 - b^2
\end{equation}
```

Ce qui donne le visuel suivant :

$$(a+b)(a-b) = a^2 - ab + ab - b^2 \tag{3.2}$$

$$(a+b)(a-b) = a^2 - b^2 \tag{3.3}$$

Nous pouvons alors voir que le résultat obtenu n'est pas forcément très esthétique. En effet, il serait préférable que l'interligne entre les deux équations soit réduit, que les signes égal soient alignés et même que l'on ne mette rien à la gauche du égal de la deuxième ligne. Pour réaliser cela c'est très simple, il suffit d'utiliser une variante de l'environnement equation, à savoir l'environnement eqnarray. Le plus simple est de reprendre l'exemple précédent :

```
\begin{eqnarray}
(a+b)(a-b) &=& a^2 -ab + ab - b^2 \\
 &=& a^2 - b^2
\end{eqnarray}
```

$$\begin{aligned} (a+b)(a-b) &= a^2 - ab + ab - b^2 && (3.4)\\ &= a^2 - b^2 && (3.5) \end{aligned}$$

Il est alors observable que l'on obtient un résultats d'un niveau esthétique tout à fait différent que précédemment. Bien entendu il est possible de réaliser une telle mise en forme sans numérotation, pour cela il suffit de mettre une étoile dans le nom de l'environnement :

```
\begin{eqnarray*}
(a+b)(a-b) &=& a^2 -ab + ab - b^2 \\
 &=& a^2 - b^2
\end{eqnarray*}
```

$$\begin{aligned}(a+b)(a-b) &= a^2 - ab + ab - b^2 \\ &= a^2 - b^2\end{aligned}$$

Ensuite l'autre point important est l'écriture de système d'équations scalaires. Pour cela, il va être nécessaire de définir dans l'environnement équation un tableau qui va permettre l'organisation du système d'équation. Le mieu étant de faire un exemple :

```
\begin{equation}
\left\lbrace \begin{array}{rcl}
0 & = & 3x+2y \\
0 & = & 7x-4y \\
\end{array} \right.
\end{equation}
```

$$\left\{ \begin{array}{rcl} 0 & = & 3x+2y \\ 0 & = & 7x-4y \end{array} \right. \tag{3.6}$$

Si l'on s'intéresse un peu à cet exemple, nous pouvons constater que la syntaxe pour aligner les équations est identique à celle des tableaux que nous avons vu dans le chapitre précédent. Ensuite, vous pouvez observer que l'environnement array servant à définir le tableau pour le placement des équations est encadré par les commandes `\left\lbrace ... \right.`. La commande `\lbrace` permet de définir l'accolade à la gauche du système d'équation, on la précède de la commande `\left` afin de faire en sorte que l'accolade "englobe" bien toutes les lignes du système d'équation. De la même manière les commandes `\right.` permettent de définir qu'il n'y a aucun symbole à la droite du système d'équation. Par exemple, si l'on réécrit le même système de la manière suivante :

```
\begin{equation}
\lbrace \begin{array}{rcl}
0 & = & 3x+2y \\
0 & = & 7x-4y \\
\end{array} .
\end{equation}
```

Cela donne :

$$\left\{ \begin{array}{rcl} 0 & = & 3x+2y \\ 0 & = & 7x-4y \end{array} \right. . \tag{3.7}$$

Attention

Il faut bien faire attention au fait que même si l'on ne souhaite pas avoir de symbole à la droite du système d'équation il est nécessaire de mettre un point dans le code source. Si vous oubliez le point, il se produira alors une erreur de compilation.

3.2.4 Matrices et vecteurs

Maintenant que vous savez comment produire des équations scalaires, nous allons voir comment rédiger des équations matricielles. Pour faire cela, c'est très simple car il n'y a qu'une seule chose à retenir et il s'agit de la méthode pour écrire des matrices. Si l'on y pense un peu, une matrice peut se décomposer en deux éléments :

— un tableau
— des délimiteurs

Ainsi avec ce que nous avons déjà vu, vous êtes capable de définir une matrice. La solution qe vous pouvez ainsi proposer est la suivante :

```
\begin{equation}
\left[ \begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
\end{array} \right]
\end{equation}
```

$$\left[\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array} \right] \tag{3.8}$$

Cependant, cette solution est relativement lourde et il est facile d'oublier une accolade entrainant ainsi une erreur au moment de la compilation. Mais comme vous pouvez vous en douter, il existe une solution plus simple en LaTeX et cette solution est la suivante :

```
\begin{equation}
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
\end{bmatrix}
\end{equation}
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \tag{3.9}$$

Bien entendu, il est possible d'utiliser d'autres délimiteurs :

pmatrix : ( )

bmatrix : [ ]

Bmatrix : { }

vmatrix : | |

Vmatrix : ‖ ‖

Il s'agit là de la seule chose qu'il est nécessaire de connaitre pour écrire des équations matricielles.

3.2.5 Le reste ...

3.3 Tracer des courbes en LaTeX

A présent, il est possible avec ce que vous venez de voir d'écrire des documents LaTeX comprenant des images, tableaux ainsi que des formules mathématiques. Il est tout à fait possible de se contenter de cela, mais dans un document scientifique il y a très souvent

des courbes et graphiques. La solution de simplicité est alors de tracer ses courbes avec un programme quelconque, excel, matlab, ... D'exporter la courbe obtenue au format png, pdf ou tout autre format d'image et d'importer cette image comme nous l'avons vu au début de ce chapitre.

Il apparait alors deux problèmes avec cette solution :

Problème esthétique du au fait que les graphiques ainsi réalisé ne seront pas réalisés selon la même charte graphique que le reste du document que l'on est en train d'écrire

Problème pratique provenant du fait que si les données des courbes changes, il est nécessaire de refaire chacune des courbes avec le logiciel que l'on a choisi pour cela et de re-exporter ces dernières au format image. Ce qui selon le nombre de courbes peut s'avérer extrèmement long.

La solution pour résoudre ces deux problèmes, est alors de tracer les courbes directement en utilisant LaTeX. L'avantage est que l'on obtient un résultat qui esthétiquement est parfaitement intégré au document que l'on rédige et de plus il est relativement simple de modifier les données d'entrées des courbes et alors il ne reste plus qu'à recompiler le document pour que ces modifications soient prises en compte.
Dans la suite nous allons distinguer deux types de courbes :

1. Les courbes représentant l'évolution d'une fonction donnée sur un intervalle donné.
2. Les courbes obtenue à partir d'une série de donné ou d'un nuage de points.

Pour tracer ces deux types de courbes il est fortement recommandé d'utiliser le package PGFplots. Il faut donc placer dans l'en-tête du document la ligne suivante :

```
\usepackage{pgfplots}
```

3.3.1 Issue d'une fonction

Pour commencer, nous allons chercher à tracer l'évolution de $\cos^2 x$ pour $x \in [-\pi; \pi]$. La solution est la suivante :

```
\pgfplotsset{width=9cm,compat=1.3}
\begin{tikzpicture}
\begin{axis}[
xlabel={$x (rad)$},
ylabel={$f(x) = \cos^2(x)$}
]
\addplot[blue,domain=-pi:pi, smooth] {cos(deg(x))^2 };
\end{axis}
\end{tikzpicture}
```

L'uilisation du package PGFplots s'avère relativement intuitive, nous allons cependant détailler commande par commande le code précédent :

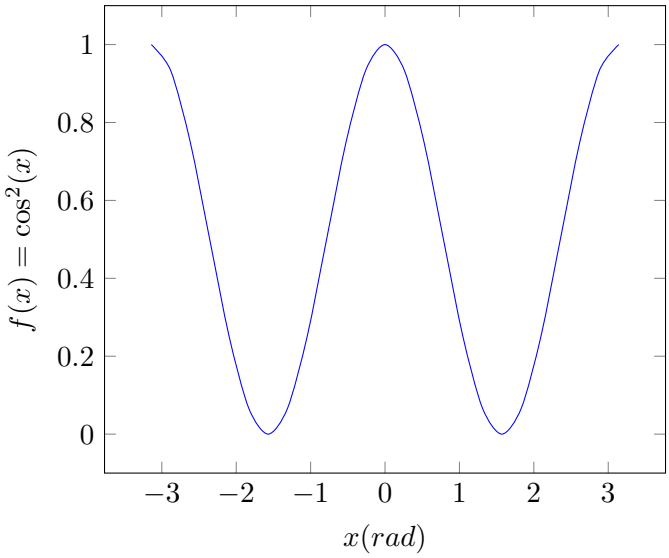
Commande 1 définie la taille du graphique

Commande 2 environnement de dessin TiKz, sera vu plus en déatil au chapitre précédent

Commande 3 définition des axes du graphique

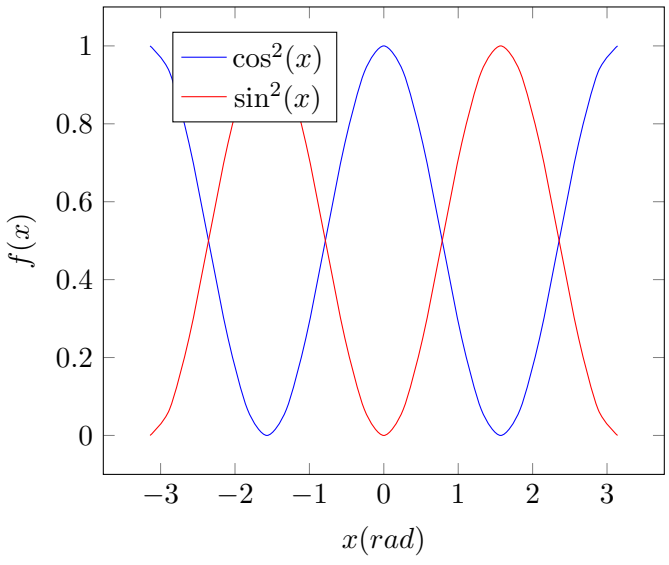
Commande 4 définition de la courbe, avec choix de la couleur, domaine d'évolution de la fonction et définition de la fonction à tracer

Le résultat alors obtenu est le suivant :



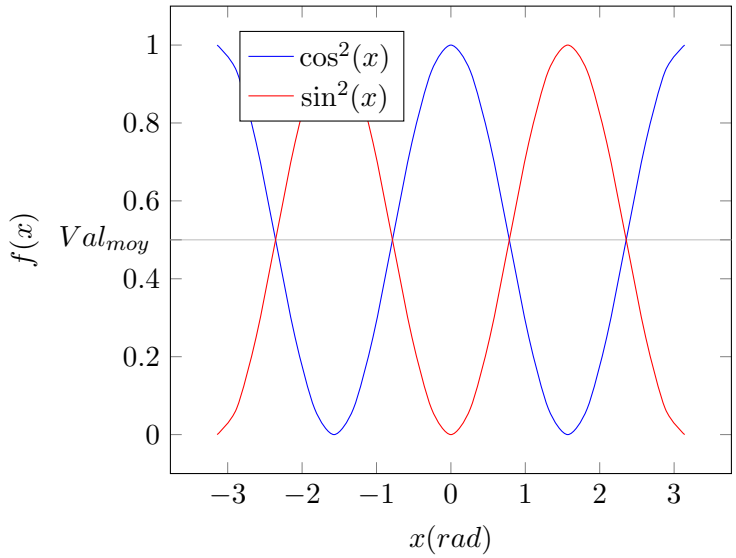
Ensuite, ce qui peut s'avérer intéressant c'est de pouvoir tracer plusieurs fonctions sur un même graphiques. C'est très simple à faire, par exemple si l'on veut tracer sur le même graphique l'évolution de $\sin^2(x)$ et $\cos^2(x)$ il suffit de procéder de la manière suivante :

```
\pgfplotsset{width=9cm,compat=1.3}
\begin{tikzpicture}
\begin{axis}[
xlabel=$x (rad)$,
ylabel={$f(x)$},
legend style={at={(0.27,0.85)}, anchor = center}
]
\addplot[blue,domain=-pi:pi, smooth] {cos(deg(x))^2 };
\addlegendentry{$\cos^2(x)$}
\addplot[red,domain=-pi:pi, smooth] {sin(deg(x))^2 };
\addlegendentry{$\sin^2(x)$}
\end{axis}
\end{tikzpicture}
```



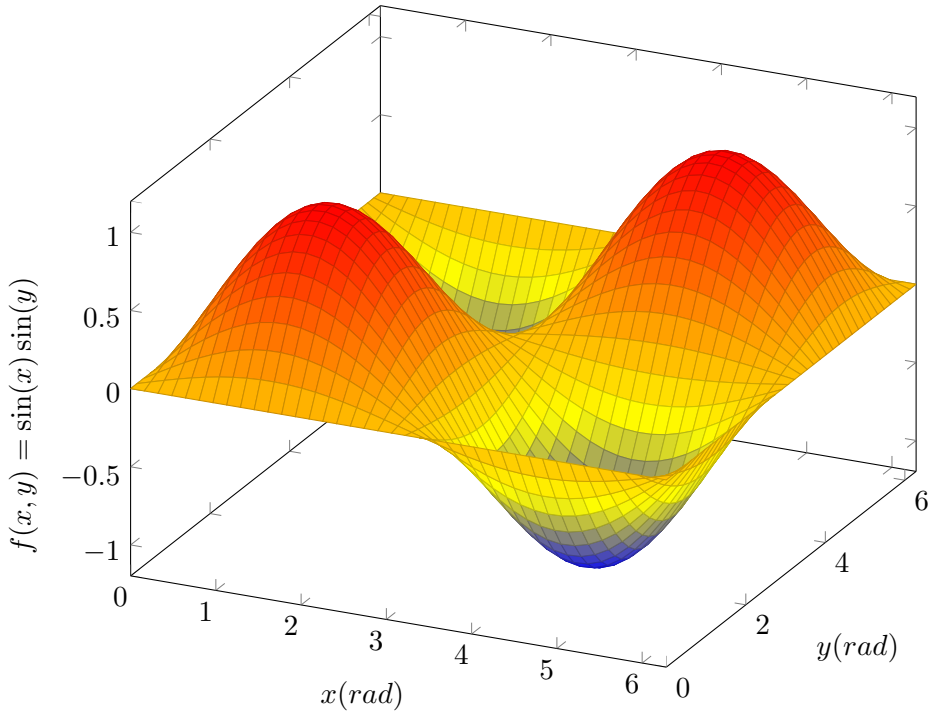
Une autre manipulation qu'il est intéressant de pouvoir réaliser, est d'ajouter des annotations sur les graphiques. Pour cela il suffit de procéder de la manière suivante :

```
\pgfplotsset{width=9cm,compat=1.3}
\begin{tikzpicture}
\begin{axis}[
xlabel=$x (rad)$,
ylabel={$f(x)$},
legend style={at={(0.27,0.85)}, anchor = center},
extra y ticks={0.5},
extra y tick labels={$Val_{moy}$ \vspace{1cm}  },
extra y tick style={grid=major, tick label style={at={(0.27,0.65)},rotate=0,anchor=east}}
]
\addplot[blue,domain=-pi:pi, smooth] {cos(deg(x))^2 };
\addlegendentry{$\cos^2(x)$}
\addplot[red,domain=-pi:pi, smooth] {sin(deg(x))^2 };
\addlegendentry{$\sin^2(x)$}
\end{axis}
\end{tikzpicture}
```



Pour finir, il est tout à fait possible de tracer des surfaces de la même manière que nous venons de le faire pour les courbes. Par exemple, si l'on veut représenter la fonction $f(x, y) = \sin(x)\sin(y)$, il suffit de procéder de la manière suivante :

```
\pgfplotsset{width=12cm,compat=1.3}
\begin{tikzpicture}
\begin{axis}[
xlabel=$x (rad)$,
ylabel={$y (rad)$},
zlabel={$f(x,y)=\sin(x)\sin(y)$}
]
\addplot3[surf,domain=0:2*pi,samples=40]
{sin(deg(x))*sin(deg(y))};
\end{axis}
\end{tikzpicture}
```



3.3.2 Issue d'un nuage de point

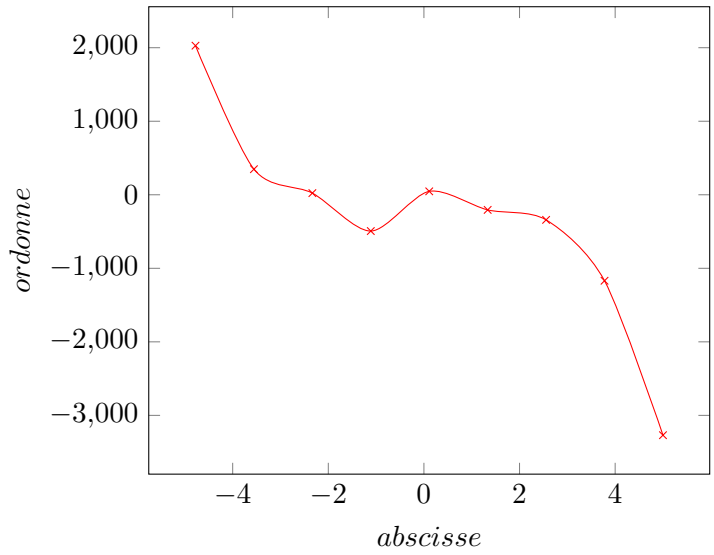
Enfin, le second type de courbes que l'on peut vouloir tracer sont des courbes issues d'une collection de points. La démarche et la syntaxe à utiliser est extrèmement proche de ce que nous venons de voir pour les courbes issues d'une fonction.
La seule différence provient du fait qu'ici il faut faire appelle à un nuage de point. Les coordonnées de ces points doivent être stockées dans un fichier `.txt` que l'on va placer dans un dossier comme nous l'avons déjà fait poiur les images. Par exemple si l'on veut tracer la courbe correspondant au point contenus dans le fichier `serie1.txt` lui même contenu dans le dossier `data` il faut procéder de la manière suivante :

```
\pgfplotsset{width=9cm,compat=1.3}
\begin{tikzpicture}
\begin{axis}[
xlabel=$abscisse$,
ylabel={$ordonne$},
]
\addplot[color=red,mark=x,smooth] file {data/serie1.txt};
\end{axis}
\end{tikzpicture}
```

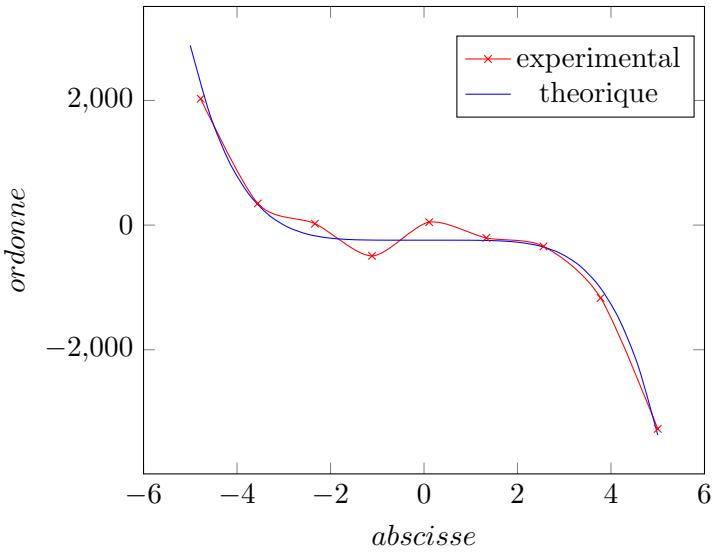
Afin d'être compréhensible par le compilateur LaTeX le fichier de donné `serie1.txt` doit être organisé de la manière suivante :

```
-4.77778 2027.60977
-3.55556 347.84069
-2.33333 22.58953
-1.11111 -493.50066
0.11111 46.66082
1.33333 -205.56286
```

```
2.55556 -341.40638
3.77778 -1169.24780
5.00000 -3269.56775
```



Bien entendu il est possible sur un même graphique de tracer des courbes issues d'une fonction mathématique et issue d'une collection de point. Cela s'avère particulièrement intéressant lorsque l'on souhaite comparer un modèle à des données expérimentales.



Pour information

La plupart du temps les séries de données que l'on souhaite utilisées sont des tableaux au format excel, il faut alors que l'on puisse passer ces fichiers excel au format `.txt` reconnu par LaTeX. Pour cela une démarche simple est la suivante :

1. Dans excel, Menu -¿ Enregistrer sous, choisir un format txt avec séparateur point-virgule par exemple. Ne surtout pas prendre la virgule comme séparateur.
2. Ouvrir avec un éditeur (par exemple notepad) le fichier ainsi sauvegardé. Utiliser la fonction de recherche et remplacement, afin de remplacer les point-virgules par un espace et les virgules des décimales par des points.

3.4 Labels et références

Nous venons de voir comment insérer des images, écrire des équations et tracer des courbes dans un document LaTeX. Ainsi nous avons à notre disposition tous les éléments nécessaires à la rédaction d'un document scientifique, ou presque. En effet, afin de rédiger un document qui soient le plus agréable possible pour le lecteur il est nécessaire de mettre en place des liens hypertexte, par exemple un sommaire "cliquable", de réaliser des liens vers les figures, ce qui permet de signifier une figure dans du texte ou bien encore de faire référence à une équation via son numéro dans du texte.

Pour réaliser cela, il est nécessaire d'appeler le package hyperref de la manière suivante :

```
\usepackage{hyperref}
```

De plus, il est nécessaire de fournir certaines indications, cela se fait à l'aide de la commande suivante :

```
\hypersetup{
    bookmarks=true,
    unicode=false,
    pdftoolbar=true,
    pdfmenubar=true,
    pdffitwindow=false,
    pdfstartview={FitH},
    pdftitle={My title},
    pdfauthor={Author},
    pdfsubject={Subject},
    pdfcreator={Creator},
    pdfproducer={Producer},
    pdfkeywords={keyword1} {key2} {key3},
    pdfnewwindow=true,
    colorlinks=true,
    linkcolor=blue,
    citecolor=green,
    filecolor=magenta,
    urlcolor=cyan
}
```

Les noms des paramètres étant tout de même relativement explicite, nous n'allons pas nous attarder sur leurs significations. Le plus important étant le choix des différentes couleurs pour les liens internes et externes, c'est à dire les quatre derniers paramètres.

3.4.1 Tables

Le premier type d'éléments que nous allons voir, sont les tables et listes. C'est à dire la table des matières, liste des figures, liste des tableaux, ... Il faut bien comprendre que ces différents éléments sont ce qu'il y a de plus important dans votre document juste après le titre de ce dernier. En effet lorsque l'on trouve un pdf sur internet par exemple, la première chose que l'on regarde c'est le sommaire et c'est pour cela que la construction du sommaire doit toujours faire l'objet d'une réflexion approfondie. Ainsi lorsque l'on trouve un document avec un sommaire clair et surtout "cliquable" permettant ainsi de regarder rapidement les éléments pouvant nous intéresser, c'est beaucoup plus agréable et cela donne envie de lire le document.

Pour générer la table des matières, c'est très simple, il suffit d'ajouter dans le code source à l'endroit où l'on veut voir apparaitre la table des matières la commande :

```
\tableofcontents
```

Il est possible de spécifier la "profondeur" de la table des matières, c'est à dire jusqu'à quel niveau (section, sous-section, sous-sous-section) l'on veut faire apparaitre les éléments. Pour cela, il suffit de placer dans l'en-tête du document la commande suivante :

```
\setcounter{tocdepth}{1}
```

Ici par exemple, dans le cas d'un document de classe report la table des matières la table des matières ne recensera que les chapitres et les sections.

Ensuite, il est possible de réaliser des listes des figures et tableaux, autrement dit des listes des deux flottant définis de base dans LaTeX. Pour ce faire, il suffit de placer dans le codes source à l'endroit où l'on veut voir ces listes apparaitres les deux commandes respectives suivantes :

```
\listoffigures
\listoftables
```

Enfin, dans la partie précédente nous avons vu comment tracer des courbes en directement en LaTeX et nous avons vu qu'il pouvait être intéressant de placer ces courbes dans un flottant défini spécialement, et ce afin de pouvoir donner un nom aux courbes que l'on trace. L'autre avantage de faire cela et que l'on a ainsi une numérotation des courbes et donc nous allons pouvoir définir une liste spécialement pour ce nouvel élément flottant. Pour rappel, nous avons défini cet élément flottant de la manière suivante :

```
\floatstyle{plain}
\newfloat{courbe}{htbp}{crb}[chapter]
\floatname{courbe}{Courbe}
```

Et donc pour générer la liste des courbes, il suffit de placer dans le corps du document la commande suivante :

```
\listof{courbe}{Liste des courbes}
```

Dans la première accolade on place le nom du flottant dont on veut établir la liste et dans la seconde accolade on met le titre que l'on veut faire apparaitre en haut de page ou sera écrite la liste ainsi générée.

3.4.2 Liens hypertextes

Nous venons de voir comment définir certains liens dans le document que l'on rédige, de manière automatique. A présent nous allons voir comment il est possible de définir d'autre liens manuellement. Pour cela nous allons diviser ces autres liens en deux catégories :

— liens internes
— liens externes

Liens internes

Par lien interne, il faut comprendre lien pointant vers un éléments du document que l'on rédige. Par exemple, on peut ainsi faire appel dans du texte à une équation écrite dix pages plus tôt. Il est possible de réaliser la même chose avec une image ou bienle titre d'une section, sous-section, ...

Pour faire cela, c'est très simple et il suffit de ne connaitre que deux commandes :

```
\label{Nom du label}
\ref{Nom du label}
```

Par exemple, pour faire référence à une équation il suffit de placer la commande `\label{Eq.1}` dans l'environnement equation contenant l'expression qui nous intéresse. Par exemple, si l'on tappe les lignes suivante :

```
\begin{equation}
f(x)=2x+4\sin\left( \frac{2\pi}{L}x \right)
\label{EqTest}
\end{equation}
```

On obtient, comme on peut s'y attendre le résultat suivant :

$$f(x) = 2x + 4\sin\left(\frac{2\pi}{L}x\right) \tag{3.10}$$

Ainsi la commande `\label{EqTest}` n'a aucune influence sur le résultat. Cependant, à présent, nous pouvons faire un "appel" à cette équation en utilisant la commande `\ref{EqTest}`. Cela nous donne alors 3.10. Nous pouvons alors remarquer une petite nuance, l'équation est numérotée avec des parenthèses alors que lors de l'appel à cette équation il n'y a pas de parenthèses. Pour corriger cela, il existe la commande `\eqref{EqTest}` qui corrige cela et fournie le résultat (3.10).

De la même manière, si l'on place la commande `\label{...}` dans un environnement flottant on peut alors faire référence à ce flottant à l'aide de la commande `\ref{...}`. Il existe également la commande `\pageref{...}` qui permet de récupérer le numéro de la page sur laquelle se situe l'élément référencé. Par exemple l'équation (3.10) se situe à la page 40.

Attention

Lorsque l'on place la commande `\label{...}` dans un flottant et bien il est impératif de la placer après la commande `\caption{...}`. Si cette condition n'est pas respectée alors l'utilisation de la commande `\ref{...}` ne renverra pas le numéro du flottant mais le numéro de la section, sous-section, ... contenant le flottant.

Liens externes

Enfin, l'autre famille de liens hypertextes que l'on peut mettre en place sont les liens externes. C'est à dire qu'il s'agit de liens pointant vers des éléments extérieures au document que l'on rédige. Par exemple, une page internet ou bien automatiser l'envoi d'un mail comme c'est le cas sur la page de garde de ce document.

Pour faire tout cela, il ne suffit que d'une seule et unique commande :

```
\href{...}{...}
```

Pour faire un lien vers une page internet, il faut procéder de la manière suivante :

```
\href{url}{description}
```

Par exemple si l'on veut faire un lien vers la page d'acceuil du site Wikipédia, il suffit de tapper la commande suivante :

```
\href{http://fr.wikipedia.org/}{page d'acceuil du site Wikipédia}
```

Ainsi cette commande nous permet d'avoir un lien vers la page d'acceuil du site Wikipédia.

Ensuite, l'autre lien que l'on peut mettre en place est l'automatisation de l'envoi de mail. Pour cela, il suffit de tapper la commande suivante :

```
\href{mailto:adresse mail}{description}
```

Par exemple, pour mettre en place le lien de la page de garde de ce document, il a suffit de tapper la commande suivante :

```
\href{mailto:basile.marchand@ens-cachan.fr}{basile.marchand@ens-cachan.fr}
```

Cela fournit alors le résultat basile.marchand@ens-cachan.fr. Bien entendu il est tout a fait possible de tapper autre chose qu'une adresse mail dans la seconde accolade de la commande. Par exemple si l'on écrit :

```
\href{mailto:basile.marchand@ens-cachan.fr}{contact}
```

Le résultat obtenu est contact.

Enfin pour finir, il est possible de faire un lien avec un document autre, par exemple faire un lien vers la présentation associée au document que l'on rédige. Pour cela, il suffit de tapper la commande :

```
\href{run:/chemin/du/fichier/}{description}
```

3.4.3 Bibliographie

Finalement, il ne reste plus qu'un seul type de lien qu'il est nécessaire de maitriser lorsque l'on rédige un document scientifique, il s'agit des références bibliographiques. En effet, si l'on souhaite rédiger un document scientifique rigoureux, il est impératif de citer ses sources.

Pour réaliser cela, il a été développé en paralèlle de LaTeX un autre outils de compilation appelé BibTeX. Le fonctionnement de BibTeX est basé sur l'utilisation d'un ou plusieur fichier `.bib` jouant le role de base de donnée des références bibliographiques. Ainsi pour faire appel à un fichier `.bib` donné il faut placer dans le corps du document la commande suivante :

```
\bibliography{/path/fichier.bib}
```

Cependant, avant même de fournir le fichier de bibliographie il faut définir le style que l'on souhaite pour la présentation de cette bibliographie. Pour cela, il suffit de tapper dans le corps du document la commande :

```
\bibliographystyle{style}
```

Le paramètre style va avoir pour principale influence la numérotation des sources, numérique ou alpha-numérique, par ordre d'apparition dans le document ou par ordre alphabétique, ... Il existe de nombreux styles possibles, cependant je vous conseil d'utiliser le style `alpha-fr`. En effet ce style permet d'avoir une numérotation des sources de type alpha-numérique avec les trois premières lettres du nom de l'auteur suivies des deux derniers chiffres de l'année de publication. Par exemple si je cite ici l'un des documents m'ayant servi pour la rdaction de ce tutoriel, on obtient [**?**]. Pour information, par défaut ne sera afficher dans les sources bibliographiques que les ouvrages ayant été cités dans le document. Pour afficher tous les ouvrages, il suffit de placer la commande `\nocite{*}`. Si l'on synthétise tout cela, pour insérer une bibliographie dans un document LaTeX, il faut ajouter dans le corps du document à l'endroit où l'on veut cette bibliographie les lignes suivantes :

```
\bibliographystyle{alpha-fr}
\bibliography{/path/fichier.bib}
```

Justement, il nous faut maintenant voir comment constituer la base de donné bibliographiques, c'est à dire comment rédiger le `fichier.bib` et ensuite comment citer un ouvrage de cette base de donné. Pour réaliser ce fichier `fichier.bib` il faut renseigner pour chacun des ouvrages que l'on souhaite citer différents champs. La nature de ces champs dépend du type de document que l'on cite, par exemple s'il s'agit d'un article dans une collection ou d'un rapport technique les rensignement à fournir ne seront pas les mêmes. Ces renseignements se présente de la manière suivante :

```
@<Type de ressource>{
<Label>,
<champ 1> = {•},
<champ 2> = {•},
<champ 3> = {•},
...
}
```

Pour définir, une ressource bibliographique il est adonc nécessaire de renseigner trois types de paramètres. Les différents types de ressources bibliographiques et les champs qui leurs sont associées sont présentés dans le tableau 3.8. Quant au parmaètre `<Label>` il s'agit de a référence qu'il faudra utiliser lorsque dans notre document on voudra préciser, à un endroit précis, qu'une ressource en particulier à été utilisée. Cela se fait à l'aide de la commande :

```
\cite{<Label>}
```

Type	**Champs descriptifs**
`article`	Un article d'une revue ou d'un magazine *Obligatoire* : `author, title, journal, year`. *Optionnel* : `volume, number, pages, month, note`.
`book`	Un livre avec un éditeur explicite *Obligatoire* : `author` ou `editor, title, publisher, year`. *Optionnel* : `volume` ou `number, series, address, edition, month, note`.
`booklet`	Une oeuvre imprimée et reliée, mais sans éditeur identifié ni institution garante *Obligatoire* : `title` *Optionnel* : `author, howpublished, address, month, year, note`
`inbook`	Une partie d'un livre, chapitre ou section ou fourchette de pages *Obligatoire* : `author` ou `editor, title, chapter` et/ou `pages, publisher, year`. *Optionnel* : `volume` ou `number, series, type, address, edition, month, note`.
`incollection`	Une partie d'un livre possédant sopn propre titre *Obligatoire* : `author, title, booktitle, publisher, year` *Optionnel* : `editor, volume` ou `number, series, type, chapter, pages, address, edition, month, note`.
`inproceedings`	Un article d'une conférence *Obligatoire* : `author, title, booktitle, year`. *Optionnel* : `editor, volume` ou `number, series, pages, address, month, organization, publisher, note`.
`manual`	Documentation technique *Obligatoire* : `title` *Optionnel* : `author, organization, address, edition, month, year, note`.
`masterthesis`	Un mémoire de master *Obligatoire* : `author, title, school, year`. *Optionnel* : `type, address, month, note`.
`misc`	Un document n'entrant dans aucunes autre catégorie *Obligatoire* : aucun *Optionnel* : `author, tittle, howpublished, month, year, note`.
`phdthesis`	Une thèse de doctorat *Obligatoire* : `author, title, school, year`. *Optionnel* : `type, address, month, note`.
`proceedings`	Actes d'une conférence *Obligatoire* : `title, year`. *Optionnel* : `editor, volume` ou `number, series, address, publisher, note, month, organization`.
`techreport`	Un rapport publié par une école ou une autre institution, d'ordinaire dans une collection *Obligatoire* : `author, title, institution, year`. *Optionnel* : `type, number, address, month, note`.

suite sur la prochaine page

Type	Champs descriptifs
`unpublished`	Un document avec un auteur et un titre mais non publié formellement *Obligatoire* : `author, title, note`. *Optionnel* : `month, year`.

TABLE 3.8 – Types de ressources bibliographiques et champs associés

La signification des différents champs descriptifs qui viennent d'être nommés est donné dans le tableau 3.9

Champ	Description
`address`	Adresse de l'éditeur ou d'une autre institution. Généralement seule la ville est nommée.
`author`	Nom(s) du/des auteur(s) du document cité.
`booktitle`	Titre du livre englobant la partie citée.
`chapter`	Numéro de chapitre ou de section, ...
`edition`	L'édition d'un livre, par exemple "*Second*". Est nécessairement un adjectif ordinal.
`editor`	Nom(s) du/des rédacteur(s) en chef. S'il existe un champ `author` alors le champ `editor` donne le directeur de la collectionb comprenant le livre.
`howpublished`	Explication d'un mode inhabituel de publication.
`institution`	Nom de l'institution présentant un rapport technique.
`journal`	Nom d'une revue.
`month`	Mois de publication de l'ouvrage ou de son écriture dans le cas d'une oeuvre `unpublished`. Pour un meilleur rendu, il est préférable d'utiliser des abréviations en trois lettres, jan, fev, mar, ...
`note`	Toute information supplémentaire aidant le lecteur.
`number`	Le numéro d'une revue, d'un magazine, d'un rapport technique, ou d'un ouvrage d'une série.
`organization`	L'organisme qui subventionne une conférence ou publie un manuel
`pages`	Un ou plusieurs numéros de pages, ou un intervalle de pages. Oar exemple, 24-112 ou 3, 18, 37-79.
`publisher`	Nom de l'éditeur.
`school`	Nom de l'établissement d'enseignement où le mémoire a été écrit.
`series`	Nom d'une collection ou d'un ensemble de livres.
`title`	Titre de l'oeuvre.
`volume`	Le volume d'un journal ou d'un livre qui en a plusieurs
`year`	Année de publication de l'ouvrage ou de son écriture dans le cas d'une oeuvre `unpublished`.

TABLE 3.9 – Signification des champs descriptifs

Pour information

A présent vous avez à votre disposition les outils nécessaires pour réaliser des bibliographies. Il vous faut cependant faire attention, en effet, la rédaction de bibliographies est soumise à des règles typologique extrèmement strictes. Ainsi, si vous souhaitez réaliser quelque chose de correcte, je ne peux que vous encouragez à regarder ce document.

Chapitre 4

Utilisation avancée de LaTeX

Résumé

Précédemment nous venons de voir tous les éléments qu'il apparait comme impératif de maitriser afin de rédiger un document scientifique en LaTeX ayant un rendu visuel qui soit le plus agréable possible.
Cependant vous vous en doutez certainement mais vous êtes loin de maitriser toutes les possibilitées qui nous sont offertes par LaTeX. Nous allons justement dans ce chapitre passer du temps sur certaines fonctionnalitées de LaTeX permettant d'améliorer encore sensiblement la qualité des documents produits.

Contents

4.1 Faire de la programmation avec LaTeX

4.1.1 Pour quoi faire ?

Dans cette première partie, nous allons voir qu'il est tout à fait possible d'utiliser LaTeX comme un langage de programmation et ainsi créé des fonctions, des structures conditionnelles, ...

La première question que l'on peut se poser est, à quoi cela va-t-t il me servir ? La réponse est très simple, cela ne sert à rien ! En effet, tout ce que je vais vous montrer ici vous pourriez tout à fait le réaliser sans faire de programmation en LaTeX. Cependant, ce que va permettre cette programmation c'est de vous simplifier la vie et quand je dis simplifier, le mot est faible.

4.1.2 Définir des commandes

L'un des éléments les plus utiles de la programmation en LaTeX est la définition de nouvelles commandes. En effet, comme vous avez pu le voir tout au long de ce document il existe de très nombreuse commandes définies à l'aide des différents packages que l'on appel. Cependant, il peut être très utile de définir ses propres commandes. Par exemple, s'il y une expression mathématique ou bien du texte qui revient souvent dans le document que l'on rédige, est bien il est pratique de définir une commande qui ecrira automatiquement cet élément répétitif. Pour définir une nouvelle commande LaTeX, il faut procéder de la manière suivante :

```
\newcommand{nom}[narg]{définition}
```

Où les différents paramètres ont la signification suivante :

nom : le nom de la commande que l'on définit. Comme tous noms de commande LaTeX , il doit nécessairement être composé de caractères alphanumériques sans espace et précédé du caractère \.

narg : le nombre d'argument que la commande va prendre, `narg` $\in [0, 9]$

définition : ensemble de commande LaTeX qui définissent notre commande.

Par exemple, pour définir une commande qui va nous faciliter la saisie d'un texte répétitif, il est possible de procéder de la manière suivante :

```
\newcommand{\test}[0]{Voici un test de commande
permettant de répéter du texte, encore et encore.}
```

Ainsi, si l'on écrit `\test \test \test \test \test` on obtient : Voici un test de commande permettant de répéter du texte, encore et encore.Voici un test de commande permettant de répéter du texte, encore et encore.Voici un test de commande permettant de répéter du texte, encore et encore.Voici un test de commande permettant de répéter du texte, encore et encore.Voici un test de commande permettant de répéter du texte, encore et encore.

De la même manière, si l'on est amené à souvent utilisé l'expression de la matrice identitée, on peut définir la commande `\Id` suivante :

```
\newcommand{\Id}[0]{%
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
}%
```

Ainsi, si l'on écrit :

```
\begin{equation}
\mathbb{I} = \Id
\end{equation}
```

on obtient le résultat suivant :

$$\mathbb{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

Pour finir, si l'on veut définir une commande avec des arguments, il faut procéder de la manière suivante :

```
\newcommand{\pres}[3]{%
Bonjour, je m'appelle #1, je suis #2 et j'ai #3 ans.
}%
```

Il suffit alors de taper la commande `\pres{Basile Marchand}{étudiant}{22}` afin d'obtenir : Bonjour, je m'appelle Basile Marchand, je suis étudiant et j'ai 22 ans.

4.1.3 Structures de contrôle

Le second point que nous allons aborder dans le cadre de la programmation en LaTeX est l'utilisation des structures de contrôles. Autrement dit, l'utilisation de la structure : `if ... then ... else ...`. L'intéret de cette struture de contrôle est le même que précédemment, c'est à dire nous simplifier la rédaction d'un document ; Par exemple, imaginons qu'il nous faille rédiger un document en deux versions ; une version confidentiel (avec les valeurs numériques, les échelles des courbes) et une version publique (aucune valeurs, pas d'échelle sur les graphiques, certaines images remplacées). Au lieu d'avoir à gérer deux documents distincts, il est plus simple de n'avoir qu'un seul et unique document LaTeX au sein duquel il va être défini des passages dont la compilation va être soumise à une condition particulière.
Tout d'abord pour utiliser cette strutcture de controle, il est nécessaire d'appeler le package `ifthen` à l'aide, comme d'habitude, de la commande suivante :

```
\usepackage{ifthen}
```

La syntaxe de la structure de contrôle en `if` est la suivante :

```
\ifthenelse{condition}{code si vrai}{code si faux}
```

Par exemple dans le cas qui nous intéresse ici, c'est à dire l'édition de texte suivant la valeur d'un paramètre, le plus simple est de définir un paramètre booléen et alors la condition de la structure de controle consistera à regarder si ce booléen est vrai ou faux. Pour définir, un paramètre booléen, la commande à utiliser est la suivante :

```
\newboolean{nom}
```

Après avoir défini ce paramètre booléen, il faut lui assigner une valeur, vrai ou faux. Pour faire cela, il suffit de taper la commande suivante :

```
\setboolean{nom}{valeur}
```

Où le paramètre `valeur` est remplacé soit par `true` soit par `false`. Pour plus de facilité d'utilisation, je ne peux que vous conseiller de définir le paramètre booléen dans les premières lignes de l'en-tête du document que vous êtes en train de rédiger. Ainsi, il ne reste plus qu'à placer dans les sources du document que l'on rédige la structure conditionnelle là où l'on en a besoin.

```
\ifthenelse{\boolean{nom}}{texte à écrire et commande à executer si
le booléen est vrai}{texte à écrire et commande à executer si le
booléen est faux}
```

Vous remarquerez que pour faire comprendre à la fonction `ifthenelse` qu'elle doit regarder la valeur du paramètre booléen il ne faut pas juste lui fournir le nom de ce paramètre mais également lui préciser qu'elle a à faire à un booléen à laide de la commande `boolean{nom}`. Par exemple, si l'on défini un paramètre `faim` à l'aide de la commande suivante :

```
\newboolean{faim}
```

On définit ensuite la structure de contrôle suivante :

```
\ifthenelse{\boolean{faim}}{J'ai vraiment très faim mais la quiche n'est
pas encore prête}{Ce n'est pas grave que la quiche ne soit pas cuite
car je n'ai pas faim}
```

Si da ns un premier temps on défini le paramètre `faim` comme étant vrai, à l'aide de la commande `\setboolean{faim}{true}` et bien à l'emplacement de la structure de contrôle il va apparaitre le texte, J'ai vraiment très faim mais la quiche n'est pas encore prête. Alors que si l'on défini le paramètre `faim` comme étant faux, et bien à la place de la même structure de controle que précédemment c'est le texte, Ce n'est pas grave que la quiche ne soit pas cuite car je n'ai pas faim, qui s'affichera.

4.2 Faire des dessins avec TikZ

4.2.1 TikZ quésaco ?

Dans le chapitre précédent, nous avons vu à peu près tout ce qu'il est nécessaire de connaitre pour rédiger un ocument en LaTeX. Cependant, vous l'aurez peut être remarqué mais il y a tout de même un manque. En effet, nous avons vu comment inclure des images dans un document mais cela nécesite que ces images existe, il est donc nécessaire de faire ces dessins avec un logiciel de DAO [1] et ensuite importer ce dernier dans notre document

1. DAO : Dessin Assisté par Ordinateur, les deux logiciels de DAO les plus connues sont Adobe Illustrator et Inkscape.

LaTeX. Le problème de cette solution est que la modification d'une image n'est pas simple, elle nécessite de repasser par le logiciel de DAO que l'on a utilisé. Comme nous l'avons vu pour tracer des courbes, la solution la plus efficace serait donc de réaliser les dessins intégralement sous LaTeX. Pour réaliser cela, il existe plusieurs solutions, cependant nous n'allons ici ne nous intéresser qu'à une seule, PGF/Tikz.

Il vous faut savoir que PGF/Tikz est un outil de dessin très poussé qui, avec du temps et de l'entrainement et encore du temps ..., peut permettre de réaliser des images très complexes et de grandes qualités. Cependant, étant donné la complexité de ce travail nous ne pouvons pas ici présenter toutes les possibilités de PGF/Tikz, nous n'allons nous intéresser qu'à certains éléments de ce module. En revanche si vous êtes intéressé par la maitrise complète de cet outil, je ne peux que vous encourager fortement à aller lire le livre

4.2.2 Formes élémentaires

4.2.3 Arbres et organigrammes

4.3 Algorithme

4.3.1 Intérêt

A présent, nous allons nous pencher sur l'utilisation du package `algorithm2e`. Ce package ne va s'avérer utile que pour les personnes réalisant des algorithmes informatiques. Si la chose a plus poussée en informatique que vous puissiez faire est de rédiger un document en LaTeX et bien cela ne vous servira pas énormément, on peut même dire que cela s'avèrera inutile, de lire cette section.

L'intérêt principal du package `algorithm2e` est qu'il va nous permettre de rédiger du pseudo-code. C'est à dire que l'on va pouvoir rédiger des algorithmes de calculs avec une syntaxe proche de celle utilisé par les langages informatique, en particulier le langage C.

4.3.2 Mise en oeuvre

4.3.3 Exemple

4.4 Faire des présentations avec Beamer

4.4.1 Beamer ?

Pour finir, nous allons rapidement nous intéresser à Beamer. Tout d'abord beamer est une classe de document LaTeX qui permet de réaliser des présentations type PowerPoint. Tout d'abord, je ne vais pas vous mentir, réaliser ses présentations en beamer est une pratique extrèmement chronophage. Quel est l'intéret alors me direz vous, et bien c'est exactement le même que celui qui nous pousse à écrire nos document en LaTeX plutôt qu'en Word, c'est à dire le fait que l'on ne se concentre que sur le fond et pas sur la forme lorsque l'on rédige. De plus, étant donné les règles "strictes" de LaTeX il n'y a que peut de risque à se retrouver avecune présentation carnaval, comme on en voit trop souvent, c'est à dire une présentation avec des animations dans tous les sens.

4.4.2 Utilisation de Beamer

Chapitre 5

Pour aller plus loin

Résumé

A présent je pense que vous avez entre les mains tout ce qu'il vous faut savoir afin de vous lancer de vous même dans le monde du LaTeX sans être guidé. Je vais juste dans ce dernier chapitre vous donner quelques liens utiles lorsque l'on recherche des informations sur LaTeX. Et enfin je vous présenterai quelques réalisation "atypiques" faites en LaTeX afin que vous puissiez vous rendre compte des inombrables possibilitées qui vous sont offertes avec bien entendu un peu de travail et "un peu" de temps.

Contents

5.1 Rechercher des informations sur LaTeX

5.2 Obtenir des modèles de documents

5.3 Autres possibilités de LaTeX

Annexes

Fichier style