

L<sup>A</sup>T<sub>E</sub>X

Edition technique et scientifique

Fabienne JÉZÉQUEL

Université Paris 2  
Master 2 Économie Managériale et Industrielle  
Expertise Economique & Juridique des Systèmes d'Information



# Table des matières

<b>1</b>	<b>Première utilisation de TeXnicCenter</b>	<b>5</b>
<b>2</b>	<b>Création d'un document</b>	<b>5</b>
<b>3</b>	<b>Utilisation des accents</b>	<b>5</b>
<b>4</b>	<b>Premiers éléments de mise en forme</b>	<b>6</b>
4.1	Centrer un paragraphe . . . . .	6
4.2	Mise en forme de caractères . . . . .	6
4.3	Taille des caractères . . . . .	6
4.4	Espace entre paragraphes . . . . .	6
4.5	Exercice : mise en forme d'un document court . . . . .	7
<b>5</b>	<b>Les listes</b>	<b>7</b>
5.1	Format de base . . . . .	7
5.2	L'environnement <code>itemize</code> . . . . .	7
5.3	L'environnement <code>enumerate</code> . . . . .	8
5.4	L'environnement <code>description</code> . . . . .	8
5.5	Listes imbriquées . . . . .	8
5.6	Exercice : création de listes . . . . .	9
<b>6</b>	<b>Notes de pied de page</b>	<b>9</b>
6.1	Exemple . . . . .	9
6.2	Exercice . . . . .	9
<b>7</b>	<b>Inclusion de figures</b>	<b>10</b>
7.1	Principe . . . . .	10
7.2	Exercice . . . . .	11
<b>8</b>	<b>Inclusion de tableaux</b>	<b>11</b>
8.1	Principe . . . . .	11
8.2	Fusion de colonnes . . . . .	12
8.3	Exercice . . . . .	13
<b>9</b>	<b>Les mathématiques</b>	<b>13</b>
9.1	Les différents environnements . . . . .	13
9.2	Les matrices . . . . .	13
9.3	Insertion de texte dans une formule mathématique . . . . .	14
9.4	Exercice . . . . .	14
<b>10</b>	<b>Mise en forme d'articles</b>	<b>14</b>
10.1	Compilation d'un article scientifique . . . . .	14
10.2	Utilisation du style d'un éditeur . . . . .	15

<b>11 Dessins avec TikZ</b>	<b>16</b>
11.1 L'environnement tikzpicture . . . . .	16
11.2 Chemin . . . . .	16
11.3 Traçage de ligne . . . . .	17
11.4 Surface remplie . . . . .	17
11.5 Représentation graphique de données . . . . .	17
11.6 Représentation graphique de fonctions . . . . .	18
11.7 Utilisation des nœuds . . . . .	18
11.7.1 Principe . . . . .	18
11.7.2 Exemple 1 . . . . .	19
11.7.3 Exemple 2 . . . . .	19
11.7.4 Exemple 3 : nœuds positionnés de manière absolue . . . . .	20
11.7.5 Exemple 4 : nœuds positionnés de manière relative . . . . .	20
11.8 Exercices . . . . .	21
11.8.1 Exercice 1 . . . . .	21
11.8.2 Exercice 2 . . . . .	21
<b>12 Présentations avec beamer</b>	<b>22</b>
12.1 Principe de beamer . . . . .	22
12.2 Animer une présentation . . . . .	22
12.2.1 La commande <i>pause</i> . . . . .	22
12.2.2 Les commandes de type <i>overlay</i> . . . . .	23
12.3 Exercice . . . . .	24
<b>13 Bibliographie</b>	<b>25</b>

## 1 Première utilisation de TeXnicCenter

Lors de sa première utilisation, TeXnicCenter doit être configuré.  
Une fenêtre apparaît avec pour titre *Configuration Wizard - Welcome*  
⇒ Cliquer sur *suivant*.

*Enter the full path of the directory where the executables of your TeX-distribution are located*  
⇒ C : \Program Files \MiKTeX 2.9 \miktex \bin  
⇒ Cliquer sur *suivant*.

*PostScript-viewer to use*  
⇒ C : \Program Files \gs \gs8.53 \bin \gswin32.exe  
⇒ Cliquer sur *Terminer*.

## 2 Création d'un document

Création d'un nouveau fichier : *File/New*  
Tapez le texte ci-dessous

```
\documentclass{article}
\begin{document}
Salut tout le monde !
\end{document}
```

Enregistrez le fichier sous le nom *ex1* : *File/Save As*. L'extension *.tex* est ajoutée automatiquement :  
vous avez créé le fichier *ex1.tex*.

Sélectionnez le format du fichier à générer.  
Au choix :

```
LaTeX => DVI
LaTeX => PDF
LaTeX => PS (PostScript)
LaTeX => PS => PDF
```

Génération du fichier à visualiser : *Build/Current File/Build*  
Génération et visualisation du fichier : *Build/Current File/Build and View*

## 3 Utilisation des accents

Modifiez le fichier *ex1.tex*, comme suit, afin de permettre l'utilisation des caractères accentués.

```
\documentclass{article}
\usepackage[latin1]{inputenc}
\begin{document}
Salut tout le monde !
Nous sommes à l'université Paris 2.
\end{document}
```

Visualisez le résultat. Ajoutez un saut de ligne entre les deux phrases (touche *Entrée* du clavier ou `\\` entre les deux phrases). Visualisez à nouveau le résultat.

## 4 Premiers éléments de mise en forme

### 4.1 Centrer un paragraphe

Utiliser l'icône adéquate ou les balises suivantes.

```
\begin{center}  
texte centré  
\end{center}
```

### 4.2 Mise en forme de caractères

Exemple :

```
\textbf{texte en gras} ou {\bf texte gras}\  
\textit{texte en italique} ou {\it texte en italique}\  
\underline{texte souligné}
```

va générer :

**texte en gras** ou **texte gras**  
*texte en italique* ou *texte en italique*  
texte souligné

Certaines balises peuvent être générées par les icônes adéquates de TeXnicCenter.

### 4.3 Taille des caractères

Exemple :

```
{\tiny petit} {\normalsize normal } {\large grand} normal à nouveau
```

va générer :

petit normal grand normal à nouveau

Ces balises peuvent être générées par le menu *Format/Font Size*.

### 4.4 Espace entre paragraphes

texte 0

texte 1

```
\smallskip
```

texte 2

```
\medskip
```

texte 3

```
\bigskip
```

texte 4

va générer :  
texte 0  
texte 1  
texte 2  
texte 3  
texte 4

#### 4.5 Exercice : mise en forme d'un document court

Modifiez le fichier `forme.tex` afin d'aboutir au résultat suivant.

---

### Introduction à LaTeX

**LaTeX** est un langage permettant de créer des documents écrits (rapports, comptes-rendus, livres, lettres) de haute qualité. En effet, même si sa mise en place peut sembler rebutante au début, LaTeX permet d'obtenir un meilleur rendu final (en terme de mise en page) qu'avec les traitements de texte classiques.

Cela est lié au principe de LaTeX : l'utilisateur n'édite pas directement le document final (à la manière de *Microsoft Word* par exemple), mais il crée un fichier source qui sera compilé.

Ainsi, il peut avoir un contrôle total et rigoureux sur la mise en page. De plus, le code des fichiers LaTeX est réutilisable pour créer de nouveaux documents plus rapidement, d'où un gain de temps.

---

## 5 Les listes

### 5.1 Format de base

Toutes les listes suivent le format de base :

```
\begin{type_de_liste}
  \item Premièrement
  \item Deuxièmement
  \item Troisièmement etc.
\end{type_de_liste}
```

### 5.2 L'environnement `itemize`

Cet environnement permet d'afficher des listes à puces.

```
\begin{itemize}
  \item Premier élément
  \item Deuxième élément
  \item Troisième élément etc.
\end{itemize}
```

va générer :

- Premier élément
- Deuxième élément
- Troisième élément etc.

### 5.3 L'environnement `enumerate`

Cet environnement est conçu pour représenter des listes numérotées, dans lesquelles chaque élément est numéroté successivement.

```
\begin{enumerate}
  \item Premier élément
  \item Deuxième élément
  \item Troisième élément etc.
\end{enumerate}
```

va générer :

1. Premier élément
2. Deuxième élément
3. Troisième élément etc.

### 5.4 L'environnement `description`

L'environnement de description est légèrement différent. Vous pouvez donner une étiquette à un élément de la liste en la passant comme paramètre facultatif (bien que facultatif, il paraîtrait curieux de ne pas l'inclure!). Cet environnement est idéal pour une série de définitions, comme on en trouve dans un glossaire par exemple.

```
\begin{description}
  \item[Premier] Le premier élément
  \item[Deuxième] Le deuxième élément
  \item[Troisième] Le troisième élément
\end{description}
```

va générer :

**Premier** Le premier élément  
**Deuxième** Le deuxième élément  
**Troisième** Le troisième élément

### 5.5 Listes imbriquées

LaTeX vous permet heureusement d'insérer un environnement de liste dans une autre liste existante (jusqu'à une profondeur de quatre). Il suffit simplement de créer un environnement de liste approprié à l'endroit désiré d'une liste donnée. LaTeX se chargera de la disposition des éléments, et d'une quelconque numérotation pour vous.

```
\begin{enumerate}
  \item Le premier élément
  \begin{enumerate}
    \item Premier élément de la liste imbriquée
    \item Deuxième élément de la liste imbriquée
  \end{enumerate}
  \item Le second élément
  \item Le troisième élément
\end{enumerate}
```



va générer :

1. Le premier élément
  - (a) Premier élément de la liste imbriquée
  - (b) Deuxième élément de la liste imbriquée
2. Le second élément
3. Le troisième élément

## 5.6 Exercice : création de listes

Modifiez le fichier libre.tex afin d'aboutir au résultat suivant.

---

### Les libertés offertes par le logiciel libre :

- la liberté d'utiliser le logiciel, pour quelque usage que ce soit ;
- la liberté d'étudier le fonctionnement du programme, et de l'adapter à vos propres besoins ;
- la liberté de redistribuer des copies de façon à pouvoir aider votre voisin ;
- la liberté d'améliorer le programme, et de diffuser vos améliorations au public.

### Exemples de logiciels libres :

**Linux** c'est l'un des composants les plus importants du système GNU puisqu'il s'agit du noyau ;

**Apache** le serveur web le plus utilisé sur le marché ;

**Gimp** logiciel de création numérique en mode bitmap ;

**PostgreSQL** base de données relationnelle.

---

## 6 Notes de pied de page

### 6.1 Exemple

La création d'une note de pied de page est d'une grande simplicité\footnote{Voici un exemple de note de pied de page.}.

va générer :

La création d'une note de pied de page est d'une grande simplicité<sup>1</sup>.

### 6.2 Exercice

Dans le fichier libre.tex, ajoutez après "système GNU" la note de pied de page suivante :

GNU est un système d'exploitation libre lancé en 1984 par Richard Stallman et maintenu par le projet GNU.

---

1. Voici un exemple de note de pied de page.

## 7 Inclusion de figures

### 7.1 Principe

L'inclusion d'images nécessite l'utilisation de l'extension `graphicx` :

```
\usepackage{graphicx}
```

Pour inclure une image, il faut utiliser la commande suivante :

```
\includegraphics{nom du fichier}
```

On peut distinguer deux cas.

- Si vous souhaitez obtenir un fichier PDF, vous ne pouvez inclure que des images PNG (extension `.png`), JPEG (extension `.jpg` ou `.jpeg`) ou des fichiers PDF (extension `.pdf`) mais pas des images PostScript.
- Si vous souhaitez obtenir un fichier PostScript, vous ne pouvez inclure que des images PostScript (avec l'extension `.ps` ou `.eps`).

Exemple :

La boîte en 3D est représentée en figure `\ref{fig:box}`.

```
\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{box3D.pdf}
\caption{boîte en 3D}
\label{fig:box}
\end{figure}
```

va générer :

La boîte en 3D est représentée en figure 1.

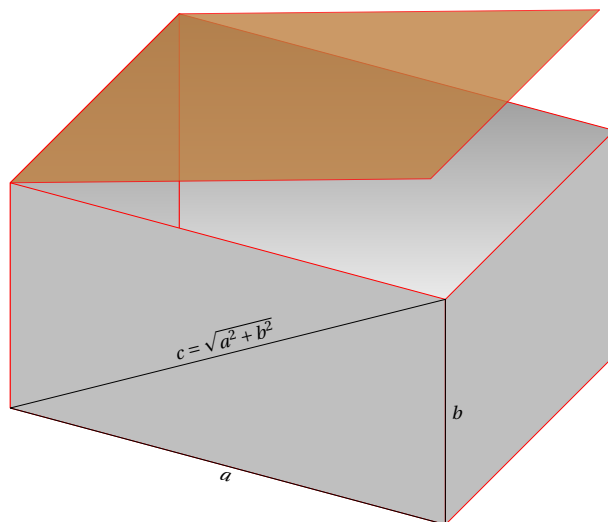


FIGURE 1 – boîte en 3D

Les instructions pour l'inclusion de figures peuvent être générées grâce au bouton adéquat de TeXnicCenter.

Pour mettre à jour les références aux figures, générer deux fois le document (commande *Build* dans TeXnicCenter).

## 7.2 Exercice

A la fin du fichier `libre.tex`, ajoutez comme suit la figure `logoGNU.pdf`. La numérotation de la figure sera différente, puisqu'il s'agit de la 1e figure du fichier `libre.tex`.

---

Le logo du projet GNU est présenté en figure 2.



FIGURE 2 – logo du projet GNU, un gnou

---

## 8 Inclusion de tableaux

### 8.1 Principe

L'environnement de base pour les tableaux en LaTeX est *tabular*.

```
\begin{tabular}{lll}
  colonne 1 & colonne 2 & colonne 3 \\
  1.1 & 1.2 & 1.3 \\
  2.1 & 2.2 & 2.3 \\
\end{tabular}
```

va générer :

colonne 1	colonne 2	colonne 3
1.1	1.2	1.3
2.1	2.2	2.3

```
\begin{tabular}{|l|c|r|}
  \hline
  colonne 1 & colonne 2 & colonne 3 \\
  \hline
  1.1 & 1.2 & 1.3 \\
  2.1 & 2.2 & 2.3 \\
  \hline
\end{tabular}
```

va générer :

colonne 1	colonne 2	colonne 3
1.1	1.2	1.3
2.1	2.2	2.3

Après `\begin{tabular}`, on indique :

- un `l` pour une colonne alignée à gauche (`left`);
- un `r` pour une colonne alignée à droite (`right`);
- un `c` pour une colonne centrée (`center`);
- un `p{largeur}` pour une colonne avec un paragraphe en pavé (c'est-à-dire justifié, mais sans alinéa); la largeur est indiquée de manière classique, par exemple `p{3cm}` ou `p{10em}`.

Les lignes sont représentées par les valeurs des colonnes séparées par des esperluettes `&` et terminées par deux contre-obliques `\\`.

Si l'on veut séparer les colonnes par des filets (traits) verticaux, on ajoutera des tubes `|` à l'endroit souhaité dans l'argument de l'environnement.

On peut aussi ajouter des filets horizontaux avec la commande `\hline`.

Le bouton *Tabular* de TeXnicCenter permet de préparer un tableau en insérant automatiquement des instructions comme celles-ci :

```
\begin{table}[htbp]
\centering
\begin{tabular}

\end{tabular}
\caption{résultats}
\label{tab:résultats}
\end{table}
```

Il ne reste plus qu'à créer le tableau.

*caption* désigne la légende du tableau.

*label* désigne la référence au tableau. Dans cet exemple, on peut obtenir la référence (le numéro) du tableau par `\ref{tab:résultats}`.

## 8.2 Fusion de colonnes

Pour fusionner des colonnes, on utilise l'instruction `\multicolumn`. Celle-ci se place à l'endroit où se trouve la première cellule. Sa syntaxe est `\multicolumn{nombre}{colonne}{texte}`, où :

- *nombre* est le nombre de colonnes fusionnées;
- *colonne* indique l'alignement (`l`, `c`, `r`).

Par exemple :

```
\begin{tabular}{|l|c|r|}
\hline
colonne 1 & \multicolumn{2}{c|}{colonnes 2 & 3} \\
\hline
1.1 & 1.2 & 1.3 \\
\hline
2.1 & 2.2 & 2.3 \\
\hline
\end{tabular}
```

va générer :

colonne 1	colonnes 2 & 3	
1.1	1.2	1.3
2.1	2.2	2.3

### 8.3 Exercice

1. Modifiez le fichier `prix.tex` afin d'aboutir au résultat suivant.

N.B. : Le symbole € s'obtient par la commande `\euro`.

---

Les prix correspondant à votre commande de matériel informatique sont détaillés dans le tableau 1.

Désignation	Prix HT	Prix TTC
Processeur	160,00 €	191,36 €
Ventilateur	20,00 €	23,92 €
Carte mère	163,00 €	194,95 €
Carte vidéo	230,00 €	275,08 €

TABLE 1 – Prix des différents composants de l'ordinateur

---

2. Modifiez ainsi le tableau du fichier `prix.tex`.

Désignation	Prix	
	HT	TTC
Processeur	160,00 €	191,36 €
Ventilateur	20,00 €	23,92 €
Carte mère	163,00 €	194,95 €
Carte vidéo	230,00 €	275,08 €

## 9 Les mathématiques

### 9.1 Les différents environnements

- formules en ligne : `$ ... $`
- formules centrées : `\[ ... \]` ou `\begin{displaymath}`  
...  
`\end{displaymath}`
- formules centrées numérotées : `\begin{equation}`  
...  
`\end{equation}`

### 9.2 Les matrices

À condition d'insérer `\usepackage{amsmath}` plusieurs environnements peuvent être utilisés pour l'écriture de matrices :

- `matrix` : matrice sans délimiteur
- `pmatrix` : matrice entre parenthèses
- `vmatrix` : matrice entre barres
- `Vmatrix` : matrice entre doubles barres
- `bmatrix` : matrice entre crochets
- `Bmatrix` : matrice entre accolades.

Exemple :

```
\[
\begin{pmatrix}
a_1 & b_1 \\
a_2 & b_2
\end{pmatrix}
\]
```

va générer :

$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}$$

### 9.3 Insertion de texte dans une formule mathématique

Exemple :

```
\[a > b \ \{\rm et \} \ c\geq d \]
```

va générer :

$$a > b \text{ et } c \geq d$$

Pour l'écriture de formules mathématiques, on pourra se référer à <http://fr.wikibooks.org/wiki/LaTeX>

### 9.4 Exercice

Reproduisez les textes et formules ci-après.

1. On considère le système linéaire  $Ax = b$  avec

$$A = \begin{pmatrix} -3 & 3 & 6 \\ 1 & 2 & 2 \\ -2 & 1 & 4 \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} -3 \\ 2 \\ -1 \end{pmatrix}.$$

2.  $I_g = \int_0^1 \int_0^1 g(x, y) \, dx dy = \int_0^1 \int_0^1 x^2 y^2 \arctan(x) \, dx dy = \frac{\pi - 2 + 2 \log(2)}{36}$ .
3. Let  $f_j = \int_{jL}^{(j+1)L} \phi(x) dx$  and  $g_m = \sum_{j=0}^m f_j$ ,  $\lim_{m \rightarrow \infty} g_m = g$ .
4.  $I = \int_0^1 \frac{6x^3 - 15x^2 - 28x + 22}{9x^2 + 12x + 4} dx = 1$ .
5.  $g = \int_0^\infty e^{-ax} dx = \frac{1}{a}$ , where  $a > 0$ .

## 10 Mise en forme d'articles

### 10.1 Compilation d'un article scientifique

On se propose de compiler le fichier *article\_petsc.tex* du dossier *supercomp*. Il s'agit d'un article publié dans "Journal of Supercomputing".

Chargez *article\_petsc.tex* dans TeXnicCenter.

Demandez le traitement suivant : LaTeX => PS => PDF

Puis effectuez succesivement toutes les actions suivantes :

```
Build/Current File/Build
Build/Current File/BibTeX
Build/Current File/Build
Build/Current File/Build
```

On obtient *article\_petsc.ps* (PostScript) et *article\_petsc.pdf*.

## 10.2 Utilisation du style d'un éditeur

On se propose d'utiliser le style fourni par l'éditeur Elsevier. Le dossier ELSEVIER contient trois fichiers accessibles sur le site internet de cet éditeur. Allez dans ce dossier. Chargez *template-num.tex* dans TeXnicCenter, puis générez les fichiers *template-num.ps* et *template-num.pdf*.

Le dossier GREMLINS\_ELSEVIER contient le même fichier *article\_petsc.tex* que celui du dossier *supercomp*. On se propose de mettre en forme le fichier *article\_petsc.tex* en vue de son éventuelle publication par l'éditeur Elsevier.

Allez dans le dossier GREMLINS\_ELSEVIER. Copiez-y le fichier *elsart.cls*.

Modifiez le fichier *article\_petsc.tex* du dossier GREMLINS\_ELSEVIER afin qu'il soit conforme au modèle *template-num.tex* :

1. changez la classe utilisée (initialement *article.cls*) en *elsart.cls*.
2. commentez à l'aide de % l'utilisation de *latex8*
3. commentez à l'aide de % l'utilisation de *times*
4. placez de manière adéquate les deux instructions suivantes :

```
\begin{frontmatter}
\end{frontmatter}
```

5. renseignez pour chaque auteur les blocs suivants :

```
\author{ }
\address{ }
```

Attention ! Enlevez les passages à la ligne dans le bloc *address*.

6. commentez à l'aide de % l'utilisation de *maketitle*
7. indiquez les mots-clé à l'aide des commandes suivantes :

```
\begin{keyword}
% keywords here, in the form: keyword \sep keyword
\end{keyword}
```

Ne vous préoccupez pas des codes PACS : la commande `\PACS` peut être omise.

Générez les fichiers *article\_petsc.ps* (PostScript) et *article\_petsc.pdf* (conformes au style d'Elsevier) de la même manière qu'en 10.1.

## 11 Dessins avec TikZ

### 11.1 L'environnement tikzpicture

L'environnement tikzpicture permet de déclarer à LaTeX que l'on commence une image TikZ.

```
\usepackage{tikz}
\usepackage{pgfplots} %pour tracer des courbes
\begin{document}
...
\begin{tikzpicture}
...
\end{tikzpicture}
...
\end{document}
```

Si l'on veut multiplier la taille de l'image dessinée par trois, on utilise :

```
\begin{tikzpicture}[scale=3]
...
\end{tikzpicture}
```

### 11.2 Chemin

On peut définir un point par ses coordonnées cartésiennes (x,y). Un chemin est une série de points séparés par des "opérations d'extension". Celles-ci définissent le type de trait qui relie les points :

- (deux tirets) : un segment de droite relie les deux points ;
- rectangle : trace un rectangle horizontal dont les angles sont les deux points.

Par exemple, un chemin sous la forme d'une ligne brisée sera de la forme

```
(x0,y0) -- (x1,y1) -- (x2,y2) -- (x3,y3)
```

Si l'on termine le chemin par -- cycle, il sera fermé.

Les points peuvent être définis de manière relative par rapport à un point précédent : le signe + devant des coordonnées définit un décalage, mais laisse inchangé le point de référence (le point de départ reste la référence), tandis que l'opérateur ++ considère que le point ainsi défini est la nouvelle référence pour les points suivants.

Un cercle est un chemin défini à partir de son centre et de son rayon :

```
(x0,y0) circle (rayon)
```

où rayon est une longueur indiquée comme à l'habitude sous la forme nombre et unité. Par exemple, le cercle de centre O et de rayon 1 cm est défini par :

```
(0,0) circle (1cm)
```

Pour avoir un arc de cercle, on utilise

```
(x0,y0) arc (angledébut:anglefin:rayon)
```

Par exemple, pour avoir l'arc du cercle précédent compris entre 0 et 90°, on écrit :

```
(0,0) arc (0:90:1cm)
```



### 11.3 Traçage de ligne

La commande permettant de tracer une ligne est `\draw chemin`

Exemple : `\draw (0,0) circle (1cm)`

On peut choisir la couleur du tracé : `\draw[red] (0,0) -- (1,1);`

On peut aussi ajouter des flèches aux extrémités :

`\draw[blue,->] (0,0) -- (1,1);`

`\draw[red,<->] (1,0) -- (2,1);`

### 11.4 Surface remplie

Pour avoir une surface remplie (peinte), on utilise la commande `\fill chemin`

Cette commande accepte la couleur comme paramètre. Par exemple, pour tracer un disque gris :

`\fill[gray] (0,0) circle (1cm)`

On peut remplir avec un dégradé en utilisant la commande `\shade`. On peut passer des couleurs en argument, comme indiqué ci-après.

dégradé horizontal :

`\shade[left color=couleur1, right color=couleur2] ;`

dégradé vertical :

`\shade[top color=couleur1, bottom color=couleur2] ;`

dégradé vers l'extérieur :

`\shade[inner color=couleur1, outer color=couleur2] ;`

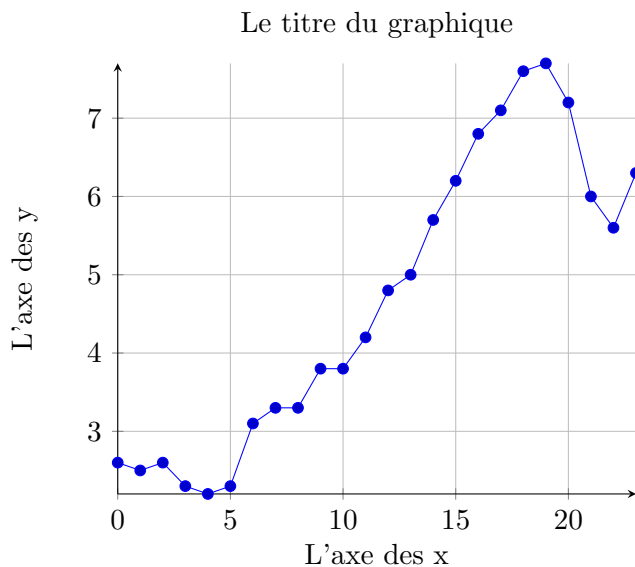
effet de balle :

`\shade[ball color=couleur].`

### 11.5 Représentation graphique de données

```
\begin{tikzpicture}
\begin{axis}[grid=major,
axis x line=bottom, axis y line = left,
xlabel={L'axe des x}, ylabel={L'axe des y},
title={Le titre du graphique}]
\addplot coordinates {
(0,2.6) (1,2.5) (2,2.6) (3,2.3) (4,2.2) (5,2.3) (6,3.1) (7,3.3) (8,3.3) (9,3.8)
(10,3.8) (11,4.2) (12,4.8) (13,5) (14,5.7) (15,6.2) (16,6.8) (17,7.1) (18,7.6)
(19,7.7) (20,7.2) (21,6) (22,5.6) (23,6.3)};
\end{axis}
\end{tikzpicture}
```

va générer :



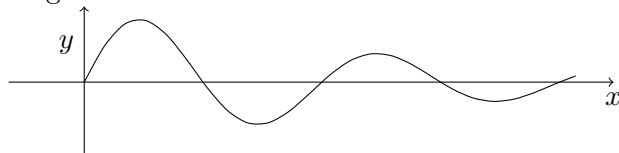
On peut agir sur la forme et les couleurs des lignes :

```
\addplot[mark=none,draw=red,dashed,ultra thick]
coordinates { . . . } ;
```

## 11.6 Représentation graphique de fonctions

```
\begin{tikzpicture}
\shorthandoff{:} %sinon conflit avec babel
\draw [->] (-1,0) -- (7,0) node [at end, below] {$x$};
\draw [->] (0,-1) -- (0,1) node [near end, left] {$y$};
\draw[smooth,domain=0:6.5] plot function{sin(2*x)*exp(-x/4)};
\shorthandon{:}
\end{tikzpicture}
```

va générer :



Les amateurs de linux compileront avec `pdflatex -shell-escape fichier.tex`

## 11.7 Utilisation des nœuds

### 11.7.1 Principe

Un nœud est une boîte qui peut contenir du texte. Pour obtenir un nœud aux coordonnées  $x$  et  $y$ , on utilise la commande `\node at (x,y) {du texte};`

La boîte du nœud peut être remplie, son bord peut être tracé et on peut choisir la couleur du texte :

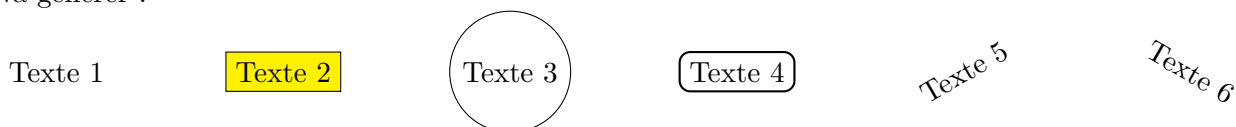
```
\node[fill=yellow,draw=black,text=blue] at (0,0) {du texte};
```

```

\begin{tikzpicture}
\node {Texte 1}; %équivalent à : \node at (0,0) {Texte 1};
\node[fill=yellow,draw=black] at (3,0) {Texte 2};
\node[circle, minimum size=1cm,draw=black] at (6,0) {Texte 3};
\node[thick,rounded corners,draw=black] at (9,0) {Texte 4};
\node[rotate=30] at (12,0) {Texte 5}; %angle en degrés
\node[rotate=-30] at (15,0) {Texte 6};
\end{tikzpicture}

```

va générer :



Par défaut, le texte est centré sur le point (x,y). Pour modifier ce comportement, node accepte des options, qui comme toutes les options LaTeX se notent entre [ ] : above (au-dessus), below (en-dessous), right (à droite), left (à gauche), et leurs combinaisons : above left (en haut à gauche), above right (en haut à droite), below left (en bas à gauche), below right (en bas à droite).

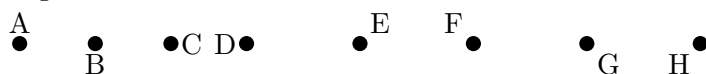
### 11.7.2 Exemple 1

```

\begin{tikzpicture}[scale=0.5]
\draw (0,0) node[above]{A}; \fill[black] (0,0) circle (0.2cm);
\draw (2,0) node[below]{B}; \fill[black] (2,0) circle (0.2cm);
\draw (4,0) node[right]{C}; \fill[black] (4,0) circle (0.2cm);
\draw (6,0) node[left]{D}; \fill[black] (6,0) circle (0.2cm);
\draw (9,0) node[above right]{E}; \fill[black] (9,0) circle (0.2cm);
\draw (12,0) node[above left]{F}; \fill[black] (12,0) circle (0.2cm);
\draw (15,0) node[below right]{G}; \fill[black] (15,0) circle (0.2cm);
\draw (18,0) node[below left]{H}; \fill[black] (18,0) circle (0.2cm);
\end{tikzpicture}

```

va générer :



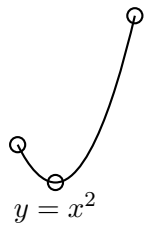
### 11.7.3 Exemple 2

```

\begin{tikzpicture}[scale=0.5]
\draw[thick] (-1,1) parabola bend (0,0) (2.1,4.41);
\draw[thick] (-1,1) circle (0.2cm);
\draw[thick] (0,0) circle (0.2cm);
\draw[thick] (2.1,4.41) circle (0.2cm);
\node[below] {$y=x^2$};
\end{tikzpicture}

```

va générer :



#### 11.7.4 Exemple 3 : nœuds positionnés de manière absolue

```
\begin{figure}
\centering
\begin{tikzpicture}[scale=0.5,text centered]
\node[text width=1.5cm,name=Fjava] at (0,0) {Fichier\\.java};
\node[draw=black,name=comp] at (6.,0) {Compilateur};
\node[text width=1.8cm,name=Fclass] at (12,0) {Fichier(s)\\.class};
\node[draw=black,name=inter] at (18,0) {Interpréteur};
\draw[->](Fjava) -- (comp);
\draw[->](comp) -- (Fclass);
\draw[->](Fclass) --(inter);
\end{tikzpicture}
\caption{Programmation en langage Java}
\end{figure}
```

va générer :

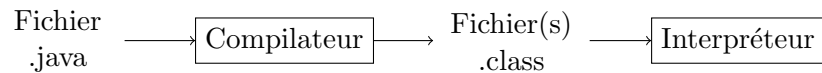


FIGURE 3 – Programmation en langage Java

Dans cette exemple, nous créons quatre nœuds nommés Fjava, comp, Fclass et inter. Ces nœuds sont ensuite reliés par des flèches.

Préciser le paramètre *text width* permet que le texte d'un nœud occupe plusieurs lignes. Les passages à la lignes sont effectués grâce à `\\`.

#### Remarque :

```
\node[draw=black,name=comp] at (6.,0) {Compilateur};
```

est équivalent à :

```
\node[draw=black] (comp) at (6.,0) {Compilateur};
```

#### 11.7.5 Exemple 4 : nœuds positionnés de manière relative

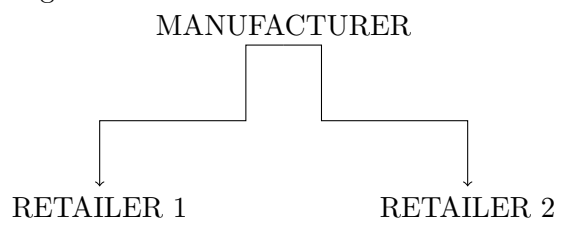
```
\begin{tikzpicture}[node distance=1cm]
\node[name=manufacturer] {MANUFACTURER};
\node[below=2cm of manufacturer] (dummy) {};
```

```

\node[left=of dummy] (retailer1) {RETAILER 1};
\node[right=of dummy] (retailer2) {RETAILER 2};
\draw[->] (manufacturer.south) -- ++(-0.5,0) -- ++(0,-1) -| (retailer1.north);
\draw[->] (manufacturer.south) -- ++(0.5,0) -- ++(0,-1) -| (retailer2.north);
\end{tikzpicture}

```

va générer :



**Remarque :** Dans le tracé des flèches, -| permet de relier deux points par une ligne à angle droit.

## 11.8 Exercices

### 11.8.1 Exercice 1

Complétez le fichier offre\_demande.tex afin qu'il contienne le graphique suivant.

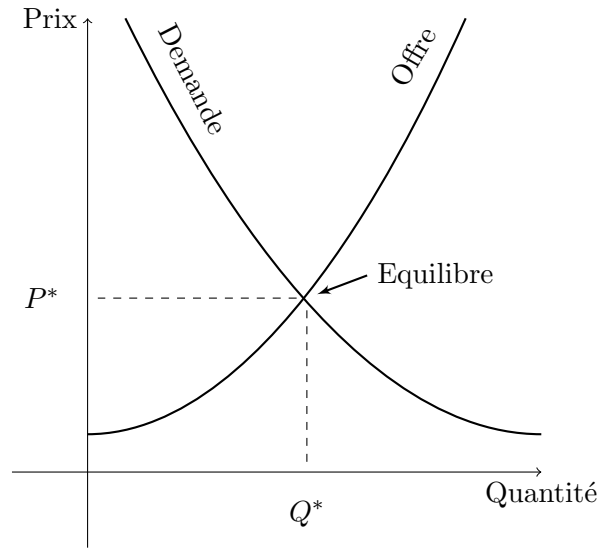


FIGURE 4 – Evolution de l'offre et de la demande

### 11.8.2 Exercice 2

Complétez le fichier organigramme.tex afin qu'il contienne l'organigramme suivant.

**Remarque :** Des options telles que **text width=...cm**, **text centered**, **minimum size=...cm** peuvent être indiquées dans les arguments de *tikzpicture*.

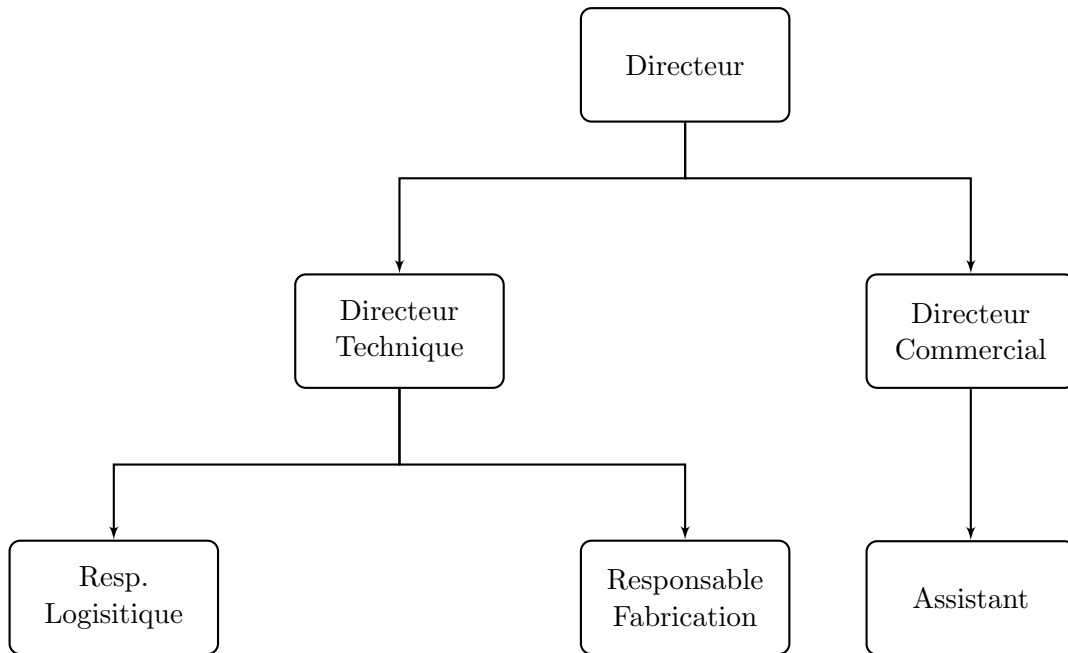


FIGURE 5 – Nouvel organigramme

## 12 Présentations avec beamer

### 12.1 Principe de beamer

Comme toutes les classes, BEAMER se charge par

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

On précise ensuite :

```
\usetheme{nom du theme}
```

```
\usecolortheme{nom du theme de couleur}
```

- Exemples de thèmes : Madrid, Antibes, Hannover, Warsaw.
- Exemples de thèmes de couleur : seahorse, rose, crane.

A partir du fichier slides.tex, générez slides.pdf. Visualisez slides.pdf.

Faites varier les thèmes et les thèmes de couleur dans slides.tex. A chaque changement, générez et visualisez slides.pdf.

Examinez en détail le fichier slides.tex.

### 12.2 Animer une présentation

#### 12.2.1 La commande *pause*

Il peut être nécessaire de ne pas faire apparaître tous les éléments d'une diapo d'un seul coup, mais de les faire apparaître un à un. La façon la plus simple, mais également la moins flexible, est d'utiliser la commande `\pause`. Son fonctionnement est très simple. Dans un premier temps n'est affiché que ce qui est placé avant la fonction `\pause`. Dans un deuxième temps, après avoir appuyé sur entrée, une flèche du clavier ou le bouton de gauche de la souris, pour passer à la diapo suivante, tout ce qui est après le `\pause` est affiché jusqu'à la fin de la diapo ou au `\pause` suivant.

Dans la dernière diapo de slides.tex, utilisez la commande `\pause` pour faire apparaître un à un les différents éléments de la liste.

### 12.2.2 Les commandes de type *overlay*

Dans la suite, on désigne par diapo l’entité définie par `\begin{frame} . . . \end{frame}`. Le terme *calque* (ou slide) fait référence à toutes ces pages créées par des commandes comme `\pause`. Une série de calques sont superposés pour fabriquer la diapo. Donc une diapo porte un numéro de page unique et peut être constituée de un ou plusieurs calques.

Il existe plusieurs commandes de type overlay, mais elles ont toutes la caractéristique de comprendre l’élément suivant `<n>` où `n` indique le ou les calques sur lesquels doit apparaître l’élément. `n` peut prendre les valeurs suivantes :

- un nombre par exemple `<2>` qui signifie que l’élément concerné n’apparaîtra que sur le calque 2 et sur aucun autre
- `<n1,n2, . . . >` par exemple `<2,5,6>` l’élément sera présent uniquement sur les calques 2, 5 et 6
- `<n->` par exemple `<2->` l’élément sera présent sur le calque 2 et tous les suivants
- `<-n>` par exemple `<-4>` l’élément sera présent uniquement sur les premiers calques et jusqu’au 4e (c’est-à-dire les calques 1, 2, 3 et 4)
- `<n1-n2>` par exemple `<3-6>` l’élément sera présent sur les calques 3 à 6 (c’est-à-dire les calques 3, 4, 5 et 6).

BEAMER reconnaît certaines commandes de L<sup>A</sup>T<sub>E</sub>X comme pouvant directement être “temporisables”. Il suffit de leur adjoindre un `<n>` :

```
\textbf, \textit, \textsl, \includegraphics, \textrm, \item, \textsf, \alert
\color \structure
```

Voici un exemple avec `\color` et `\textbf` qui vous trouverez dans la diapo 2 du fichier animation.tex.

Voici un `\textbf<2->{exemple} {\color<3>{green}d’animation}`.

A partir du fichier animation.tex, générez le fichier animation.pdf. Visualisez la diapo 2 de cette présentation.

Examinez les diapos 3 à 5 (“Une liste animée”) du fichier animation.tex. Ces diapos montrent de trois manières équivalentes comment faire apparaître un à un les éléments d’une liste. Visualisez les diapos 3 à 5 de cette présentation.

On peut mettre lors de la “temporisation”, le texte en “alert” (de couleur rouge par défaut). Examinez les diapos 6 et 7 (“Une liste animée avec alert”) du fichier animation.tex.

Le chiffre après le @ indique sur quel calque intervient la mise en “alert”. On peut utiliser des formes plus complexes comme `<alert@2->` ou `<alert@2-4,6>`.

On peut combiner l’apparition et la mise en “alert” (voir diapo 8 du fichier animation.tex).

Pour “temporiser” du texte en dehors de listes ou de commandes directement “temporisables”, on utilise les commandes `\only` et `\onslide`. Elles s’utilisent de la même façon :

```
\only<1>{le texte à temporiser}
\onslide<2>{le texte à temporiser}
```

La différence entre les deux commandes, est que `\onslide` contrairement à `\only` réserve l'espace occupé par le texte affecté à la "temporisation".

Examinez puis visualisez les diapos 9 et 10 du fichier `animation.tex`.

Par défaut, les objets absents d'un calque sont invisibles. On peut remplacer cette invisibilité par une transparence de 30% par la commande `\setbeamercovered{transparent=30}`.

Décommentez cette commande placée au début du fichier `animation.tex`.

Avec `\visible` les objets non présents sur un calque sont invisibles. Avec `\uncover` ils sont transparents avec la valeur définie par `\setbeamercovered`.

Examinez puis visualisez les diapos 11 et 12 du fichier `animation.tex`.

**Remarque :** Si on reprend la commande `\onslide` avec une transparence de 30%, elle se comporte comme `\uncover`.

### 12.3 Exercice

A partir du fichier `quadrat.tex`, générez le fichier `quadrat.pdf`.

Créez une présentation animée en utilisant le fichier `quadrat.tex`. Le fichier `quadrat_pres.pdf` est un exemple d'une telle présentation.



## 13 Bibliographie

### **L<sup>A</sup>T<sub>E</sub>X**

- L<sup>A</sup>T<sub>E</sub>X pour l' impatient, W. Appel et al, MiniMax, 2009.
- <http://fr.wikibooks.org/wiki/LaTeX>
- [http://w3.bretagne.ens-cachan.fr/math/people/gregory.vial/files/LaTeX/TP\\_LaTeX.pdf](http://w3.bretagne.ens-cachan.fr/math/people/gregory.vial/files/LaTeX/TP_LaTeX.pdf)

### **Les tableaux**

- <http://chl.be/glmf/morningstar.tuxfamily.org/magazine/latex/al/04/article.html>
- [http://fr.wikibooks.org/wiki/LaTeX/Faire\\_des\\_tableaux](http://fr.wikibooks.org/wiki/LaTeX/Faire_des_tableaux)

### **Les maths**

- [http://fr.wikibooks.org/wiki/LaTeX/\`Ecrire\\_des\\_math\`ematiques](http://fr.wikibooks.org/wiki/LaTeX/\`Ecrire_des_math\`ematiques)
- <http://fr.wikibooks.org/wiki/LaTeX>

### **TikZ**

- [http://fr.wikibooks.org/wiki/LaTeX/Dessiner\\_avec\\_LaTeX/Dessiner\\_avec\\_PGF\\_et\\_TikZ](http://fr.wikibooks.org/wiki/LaTeX/Dessiner_avec_LaTeX/Dessiner_avec_PGF_et_TikZ)
- <http://www.texample.net/tikz/examples>
- <http://bertrandmasson.free.fr/index.php?categorie6/latex-pgf-tikz>
- <http://jerome.landre.pagesperso-orange.fr/docs/tikz.pdf>
- <http://distrib-coffee.ipsl.jussieu.fr/pub/mirrors/ctan/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf> (doc complète en 726 pages)
- <http://www.tug.org/pracjourn/2007-1/mertz/mertz.pdf> (doc en 22 pages)

### **beamer**

- <http://bertrandmasson.free.fr/index.php?categorie8/beamer>
- <http://www.math-linux.com/spip.php?article76>