

# TP d'initiation à L<sup>A</sup>T<sub>E</sub>X

Ludovic GOUDENÈGE\*    Mathieu LEROY-LERÊTRE†    Grégory VIAL‡

22 mars 2010

ENS Cachan Bretagne

## Résumé

L'apprentissage de L<sup>A</sup>T<sub>E</sub>X n'est pas une tâche aisée : beaucoup d'informations existent et il est difficile de faire le tri tout seul ; de plus, un habitué des logiciels de traitement de texte sera sans doute troublé lors de ses premiers pas. L<sup>A</sup>T<sub>E</sub>X nécessite donc une initiation, que le présent document se propose de fournir sous la forme d'un TP/cours.

Qu'apporte ou non ce TP d'initiation ? Il propose des *bases* pour débiter en L<sup>A</sup>T<sub>E</sub>X et un *survol* de beaucoup de possibilités : c'est un *point de départ* avant de consulter des références plus complètes. En conséquence, ce document n'est volontairement *pas exhaustif* (sinon, il serait au moins quatre fois plus long !).

À qui s'adresse-t-il ? Ce TP d'initiation est prévu pour une séance de 4 heures sur ordinateur, destinée à des étudiants en mathématiques et en mécatronique, de niveau L3, n'ayant jamais (ou peu) utilisé L<sup>A</sup>T<sub>E</sub>X et devant réaliser un document de moyenne taille (10 à 60 pages). Il conviendra aussi à toute personne souhaitant découvrir cet outil.

Comment est-il organisé ? Le début sert de support à la séance de TP : les sections 1 et 2 permettent de savoir ce qu'est L<sup>A</sup>T<sub>E</sub>X, avant d'aborder la section 3 qui constitue le cœur de cette initiation. La section 4 présente un aperçu de fonctionnalités utiles (mise en page, inclusion d'image, flottant, graphique, présentation vidéoprojetée). Enfin, des règles typographiques et des explications volontairement éclipsées au cours du TP (éléments d'installation, présentation de quelques packages, encodage, commandes personnelles, ...) figurent en annexe.

---

\*ludovic.goudenege[@]bretagne.ens-cachan.fr.

†mathieu.leroy-leretre[@]ens-cachan.org.

‡gregory.vial[@]bretagne.ens-cachan.fr.

# 1 Présentation de L<sup>A</sup>T<sub>E</sub>X

Voyons dans cette section quelques mots introductifs pour décrire L<sup>A</sup>T<sub>E</sub>X et voir ce que signifie « faire du L<sup>A</sup>T<sub>E</sub>X ».

## 1.1 Intérêt et utilité

L<sup>A</sup>T<sub>E</sub>X, logiciel libre et gratuit, disponible sous Linux, Mac OS et Windows, est un outil de composition performant conçu pour la rédaction de documents scientifiques de qualité : il conviendra en particulier à toute personne ayant besoin de dactylographier un texte contenant des équations, des symboles, etc. Toute publication de recherche en mathématiques est faite avec L<sup>A</sup>T<sub>E</sub>X et son utilisation s'étend à d'autres disciplines : physique, informatique ou chimie par exemple. Il s'agit aussi d'un outil efficace dans l'enseignement pour rédiger des supports de cours. Son usage est donc loin d'être réservé à des spécialistes !

À la différence des traitements de texte usuels (tel MS Word), il ne s'agit pas d'un logiciel dit « WYSIWYG<sup>1</sup> » mais plutôt d'un langage qu'il faut ensuite compiler pour voir le résultat ; c'est pour cela que la prise en main est un peu plus longue.

Remarquons dès maintenant que L<sup>A</sup>T<sub>E</sub>X ne se limite pas à l'écriture d'équations : en pouvant gérer tout seul la mise en page, la création d'une table des matières et plein d'autres choses, il permet à l'utilisateur de se concentrer sur le fond (*i.e.* le contenu du texte qu'il écrit) sans se préoccuper de la forme.

## 1.2 Prise en main

Rédiger un document en L<sup>A</sup>T<sub>E</sub>X se fait en plusieurs étapes.

1. Saisie du code source : on crée un fichier avec l'extension `.tex` (que nous appelons ici `TP_init.tex`) et l'on tape son code dans un *éditeur de texte*.
2. Compilation du fichier source : c'est ici que l'on fait appel à L<sup>A</sup>T<sub>E</sub>X, qui est un programme autonome ; cela crée plusieurs fichiers, dont `TP_init.dvi`.
3. Prévisualisation du fichier DVI : on observe à l'écran `TP_init.dvi` pour voir s'il affiche bien ce qu'on voulait ; sinon, on corrige le fichier source.
4. Impression ou diffusion : on convertit son fichier au format PDF, pour l'imprimer ou le transmettre à quelqu'un.

Ces quatre étapes sont très simples, mais les outils pour les mettre en place ne portent pas les mêmes noms selon que l'on est sous un environnement Linux, Mac OS ou Windows. Nous renvoyons à l'annexe B.1.1, où des informations techniques sont résumées ; elles sont cependant inutiles pour le bon déroulement du TP.

---

1. Acronyme de *what you see is what you get*, traduisible par « ce que vous voyez est ce que vous obtenez » ; autrement dit : le texte que l'on tape est le texte qui apparaît.

## 2 Structure d'un fichier .tex

Le fichier `.tex` est créé à partir d'un éditeur de texte. Il possède en général le squelette suivant :

- la commande `\documentclass{}` qui indique quelle *classe* (ou quel *type*) de document l'on souhaite rédiger ;
- le *préambule*, où l'on indique des options, des préférences et des choix pour la compilation ;
- le *corps de document*, délimité par `\begin{document}` et `\end{document}`, où réside le texte lui-même.

Plus précisément, le préambule contient habituellement deux catégories d'informations :

**les *packages***, qui regroupent des commandes destinées à améliorer<sup>2</sup> l'utilisation de certaines fonctionnalités ou à créer d'autres commandes. Il en existe un très grand nombre et il serait illusoire de vouloir tous les décrire ici (cf. § B.3.3). Pour utiliser un package, il faut le charger ;

**les macros ou commandes personnelles (de l'utilisateur)**, une des richesses de  $\text{\LaTeX}$  que nous ne ferons qu'effleurer (cf. § B.4).

L'objectif de ce TP étant une prise en main de  $\text{\LaTeX}$ , nous nous limiterons à la classe `article`, ainsi qu'aux packages « de base » qui sont (presque) incontournables pour un document rédigé en langue française (voir la figure 1) ; ceux-ci sont expliqués en annexe B.3.1.

```
\documentclass[12pt]{article}

\usepackage[<option à choisir>]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[a4paper]{geometry}
\usepackage[frenchb]{babel}

\usepackage{amsmath, amssymb}

\begin{document}

Bienvenue en séance de TP !

\end{document}
```

Préambule

Corps de document

FIGURE 1 – Exemple de fichier source « minimal ».

---

2. Au sens rendre plus flexible ou plus facile.

## 3 Découverte de $\LaTeX$

L'objet de ce paragraphe est d'utiliser les outils de base de  $\LaTeX$  à travers des exemples, des remarques, des conseils et des exercices. L'accent est mis sur les outils mathématiques.

### 3.1 Démarrage avec $\TeX$ nicCenter

Les ordinateurs des salles informatiques de l'Antenne sont équipés de Linux et Windows : nous utiliserons Windows, avec les consignes suivantes.

0. Lancez  $\TeX$ nicCenter.
1. Recopiez l'exemple de fichier source proposé à la figure 1, en remplaçant *<option à choisir>* par `cp1252`<sup>3</sup>.

Remarquons tout de suite qu'en général, une commande s'applique à un argument parfois accompagné d'une option :

- la commande commence par un `\`,
- l'argument(s) obligatoire(s) est mis entre `{ }`,
- l'option(s) est mise entre `[ ]`.

Ainsi, `\usepackage[cp1252]{inputenc}` signifie que l'on charge un package : il s'agit de `inputenc`, utilisé avec l'option `cp1252`.

2. Enregistrez le fichier (sous le nom `TP_init.tex` par exemple).
3. Compilez (en choisissant `LaTeX => DVI` puis en cliquant sur la 2<sup>e</sup> icône à droite, intitulée « Build current file (Ctrl+F7) »).
4. Prévisualisez (en cliquant sur la 4<sup>e</sup> icône à droite, intitulée « View output (F5) ») ; cette étape ouvre une fenêtre affichant le fichier `DVI`.
5. Observez alors à l'écran que ce qui est écrit dans le corps de document est affiché.

Si le caractère accentué n'apparaît pas correctement, c'est qu'il y a un problème d'encodage : lisez alors la note de bas de page n° 3 puis changez, au choix, l'option du package `inputenc` ou l'encodage de votre éditeur (afin qu'ils coïncident tous les deux!).

L'enregistrement ci-dessus est fait une fois pour toutes : il est dorénavant inutile de sauvegarder votre fichier avant chaque compilation. De plus, si vous laissez la fenêtre affichant le fichier `DVI` ouverte, il est aussi inutile de re-cliquer sur l'icône de prévisualisation : cette fenêtre se mettra à jour dès que vous compilerez de nouveau.

6. Obtenez un PDF (en sélectionnant au choix<sup>4</sup> `LaTeX => PDF` ou `LaTeX => PS => PDF` puis en cliquant sur les mêmes icônes que ci-dessus).

**Conseil.** Utiliser systématiquement les packages de la figure 1, dont une courte description figure en annexe B.3.1.

---

3. Cela occulte un détail technique : la notion d'encodage. Plus d'explications figurent en annexe B.3.2, à lire impérativement si vous ou une personne avec laquelle vous travaillez êtes susceptibles d'utiliser un autre système d'exploitation (ce qui sera sans doute le cas...). En effet, il faudra alors écrire autre chose que `cp1252` pour pouvoir utiliser les accents du clavier sans problèmes.

4. Ce choix importe peu pour l'instant ; néanmoins, ce ne sera pas toujours le cas (cf. § B.1.2).

## 3.2 Composition simple

Nous présentons ici quelques outils de mise en forme dont on a l'habitude dans les logiciels de traitement de texte.

Commençons par deux exemples :

Voici un exemple simple pour voir comment s'utilisent, avec `\LaTeX`, la mise en `\textbf{gras}`, les formes `\textit{italique}`, `\textsc{petites capitales}` et les familles `\texttt{type machine}`, `\textsf{sans empattement}`.

Voici un exemple simple pour voir comment s'utilisent, avec `\LaTeX`, la mise en **gras**, les formes *italique*, PETITES CAPITALES et les familles `type machine`, sans empattement.

On peut aussi changer la taille des fontes en `\small petit`, `\tiny minuscule`, `\large grand`, `\huge énorme`, etc.

On peut aussi changer la taille des fontes en petit, minuscule, grand, énorme, etc.

Quelques remarques s'imposent :

- les accents, s'ils ne sont pas sur le clavier, sont codés à l'aide de `\'` ou `\^` ou `\~`, etc. (`\'e` donnera un « é »); la cédille s'obtient avec `\c{}` (`\c{C}` donnera un « Ç »);
- on n'a pas utilisé de la même manière les commandes dans les deux exemples ci-dessus : dans le premier, les commandes s'appliquent à un argument; dans le second, elles appliquent le changement à tout ce qui suit : il faut alors les isoler pour cibler leur action;
- les accolades `{}` permettent d'isoler un *groupe* afin que certains attributs ne s'appliquent pas au reste du document (`\tiny` par exemple);
- les caractères `\` et `{}` ne s'affichent pas à l'écran mais sont *interprétés* par `\LaTeX`<sup>5</sup>;
- pour passer à la ligne (c'est-à-dire débiter un nouveau paragraphe), on utilisera  *systématiquement*  la touche « entrée » du clavier deux fois<sup>6</sup>;
- on utilisera `\emph{}` pour mettre en valeur du texte, et non pas `\textit{}`.

Deux autres exemples :

Pourquoi recommander la commande dite d'`\emph{emphase}` à celle d'`\textit{italique}` ? `\textit{Parce que dans une phrase en italique, l'\emph{emphase} ressort mais pas l'\textit{italique} !}`

Pourquoi recommander la commande dite d'*emphase* à celle d'*italique*? *Parce que dans une phrase en italique, l'emphase ressort mais pas l'italique!*

5. Il existe dix de ces caractères, dits spéciaux ou réservés; cf. [short-fr] pour en voir la liste et savoir les afficher à l'écran quand même.

6. L'utilisation de la commande `\` pour cette usage est à proscrire, sauf dans l'énoncé d'un titre.

**Conseil.** Pour mettre en valeur, proscrire l’usage du souligné (`\underline{}`) et éviter celui du gras : utiliser la commande `\emph{}`.

Ce mot est `\tiny` minuscule mais tout ce qui suit aussi car il n’est pas dans un groupe.

Ce mot est minuscule mais tout ce qui suit aussi car il n’est pas dans un groupe.

### 3.3 Mathématiques 1

$\LaTeX$  est prévu pour rédiger des formules mathématiques ; il existe deux modes mathématiques :

**le mode en ligne**, utilisé pour insérer des symboles ou des formules (souvent courtes) dans une ligne de texte, au cours d’une phrase. Il s’obtient via la syntaxe  `$\dots$`  ;

**le mode hors-texte**, permettant d’écrire une formule sur une ligne à part, centrée, au moyen de la syntaxe `\[ \dots \]`<sup>7</sup>.

Considérons tout de suite des exemples.

Soit la fonction  `$f$` , définie par  `$f(x)=2x$` .

Soit la fonction  $f$ , définie par  $f(x) = 2x$ .

Soit la fonction  `$f$` , définie par `\[ f(x)=2x^2. \]`

Soit la fonction  $f$ , définie par  $f(x) = 2x^2$ .

Soit la fonction `\[ f_n : x \mapsto \alpha^2 \sqrt{\pi} + \frac{1}{2} \sum_{i=1}^n \ln x. \]`

Soit la fonction  $f_n : x \mapsto \alpha^2 \sqrt{\pi} + \frac{1}{2} \sum_{i=1}^n \ln x$ .

La fonction  `$\zeta$`  de Riemann satisfait `\[ \zeta(s) = \sum_{n=1}^{+\infty} \frac{1}{n^s} = \prod_{p \in \mathcal{P}} \frac{1}{1-p^{-s}}. \]`

La fonction  $\zeta$  de Riemann satisfait  $\zeta(s) = \sum_{n=1}^{+\infty} \frac{1}{n^s} = \prod_{p \in \mathcal{P}} \frac{1}{1-p^{-s}}$ .

Une différence fondamentale entre ces deux modes est que le mode hors-texte prend la place — si nécessaire — d’afficher les grands symboles, ce que ne fait pas le mode en ligne ; voici un exemple : « Soit  $f$  définie par  $f(x) = \sum_{i=1}^n \int_0^i f_i \left( \frac{\ln x + \alpha}{\sin x + 1} \right)$ , avec  $\alpha \in \mathbb{R}$ . » à comparer avec « Soit  $f$  définie par

$$f(x) = \sum_{i=1}^n \int_0^i f_i \left( \frac{\ln x + \alpha}{\sin x + 1} \right),$$

avec  $\alpha \in \mathbb{R}$ . ».

7. La commande  `$\dots$`  est à proscrire en  $\LaTeX$ , même si beaucoup de gens l’utilisent encore.

*Certaines habitudes ou conventions sont à prendre lorsque l'on rédige des mathématiques ; les exercices suivants ont pour but de vous y entraîner. Ils vous permettront aussi de découvrir quelques spécificités du mode mathématique par rapport au mode texte.*

À l'aide des exemples ci-dessus, des commandes proposées par `TeXnicCenter` et éventuellement d'une liste de symboles<sup>8</sup>, trouvez comment obtenir les résultats suivants ; vous ferez en priorité les exercices n<sup>os</sup> 1, 2, 3, 5, 6, 8, 10, 13, 14, 15 et 17. (On ne demande pas de reproduire l'encadrement ni les éventuels indices donnés entre parenthèses.)

*Exercice 1.* Pour  $n$  entier naturel, on pose  $u_0 = 0$  et  $u_n = u_{n-1} + n$ . Alors

$$\forall n \in \mathbb{N}^*, u_n = \frac{n(n+1)}{2} \geq 0.$$

*Exercice 2.* La formule de Stirling exprime, pour  $n$  grand, que

$$n! \sim Cn^n \sqrt{n} \exp(-n),$$

où  $C = \sqrt{2\pi}$ . Cette constante peut se calculer en utilisant la formule de Wallis, que l'on trouve grâce aux intégrales éponymes :

$$\forall n \in \mathbb{N}, I_n = \int_0^{\frac{\pi}{2}} (\sin x)^n dx.$$

*Exercice 3.* La fonction  $\Gamma: \mathbb{R}_+^* \rightarrow \mathbb{R}$ , définie par

$$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$$

et appelée "fonction Gamma (d'Euler)", généralise la factorielle. En effet,  $\forall n \in \mathbb{N}^*, \Gamma(n+1) = n!$ . On peut aussi montrer que

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi},$$

en se ramenant à l'intégrale de Gauss  $I = \int_0^{+\infty} e^{-t^2} dt$  (par changement de variables), cette dernière valant  $\frac{\sqrt{\pi}}{2}$  (par exemple en considérant le carré de  $I$  et un passage en coordonnées polaires).

Nous venons d'illustrer quelques principes de base du mode mathématique : `LATEX` adapte la taille de certains caractères selon le mode mathématique choisi ; il est aussi capable de modifier leur apparence selon notre demande : italique mathématique pour les variables, romain pour les opérateurs, ... Il connaît déjà certains opérateurs (`\cos`, `\ker`, `\sup`, `\exp`, etc.) mais pas tous ; pour le « e » de la fonction exponentielle, on code `\mathrm{e}`.

8. `LATEX` possède un très grand nombre de symboles (cf. § 3.6.3) ; pour le TP, on peut se contenter du paragraphe 3.9 de [short-fr] distribué en cours.

**Conseil.** Pour écrire un opérateur, toujours utiliser la commande prédéfinie; si elle n'existe pas, la créer de manière *adéquate*<sup>9</sup>.

<p>Il y a <math>a</math> tel que <math>a = \max x</math> pour <math>x</math> entier relatif et <math>x+2 &lt; 0</math>.</p> <p>Il y a <math>a</math> tel que <math>a = \max x</math> pour <math>x</math> entier relatif et <math>x+2 &lt; 0</math>.</p>	$\left  \begin{array}{l} \textit{bien} \rightarrow \text{ Il y a } a \text{ tel que } a = \max x \text{ pour } x \text{ entier relatif et } x + 2 < 0. \\ \textit{mal} \rightarrow \text{ Il y a } a \text{ tel que } a = \max x \text{ pour } x \text{ entier relatif et } x + 2 < 0. \end{array} \right.$
---	---

Une autre particularité importante est que le mode mathématique est un *monde à part* du mode texte. Ainsi,

- les mises en forme habituelles (gras, italique, etc.) se font au moyen d'autres commandes : `\mathrm{}` pour passer en romain, `\mathbf{}` pour mettre en gras<sup>10</sup>, etc. De plus, les lettres calligraphiques, gothiques et ajourées sont disponibles :

<p>Soit <math>f \in \mathcal{C}</math> et <math>\sigma \in \mathfrak{S}_n</math> pour <math>n \in \mathbb{N}</math>.</p>	$\left  \text{ Soit } f \in \mathcal{C} \text{ et } \sigma \in \mathfrak{S}_n \text{ pour } n \in \mathbb{N}. \right.$
--	--

- les blancs sont entièrement gérés par L<sup>A</sup>T<sub>E</sub>X; néanmoins, il a parfois du mal et il faut alors forcer les espaces : on a vu `\`, (espace fine) mais il y a aussi `\_` (espace inter-mot classique), `\quad` (1 cadratin), `\!` (espace fine négative), etc.
- les accents sont obtenus autrement ; par exemple

<p><math>\hat{x}</math>, <math>\ddot{x}</math>, <math>\vec{n}</math>, <math>\acute{m}</math>, <math>\mathring{A}</math>, <math>\bar{f}</math>, <math>\tilde{f}</math>, etc.</p> <p><math>\mathring{m}</math>, <math>\mathring{A}</math>, <math>\bar{f}</math>, <math>\tilde{f}</math>, etc.</p>	$\left  \hat{x}, \ddot{x}, \vec{n}, \acute{m}, \mathring{A}, \bar{f}, \tilde{f}, \text{ etc.} \right.$
---	--

Par contre, ces accents n'adaptent par leur taille à la largeur des caractères accentués; on dispose néanmoins de `\widehat{}`, `\widetilde{}` et de diverses flèches :

<p>L'angle <math>\hat{ABC}</math> est moins joli que <math>\widehat{ABC}</math>.</p>	$\left  \text{ L'angle } \hat{ABC} \text{ est moins joli que } \widehat{ABC}. \right.$
--	--

*Exercice 4* (avec `\mathcal{}`). Pour  $M \in \mathcal{M}_n(\mathbb{Z})$ ,

$$M \in \text{GL}_n(\mathbb{Z}) \iff \det M = \pm 1.$$

Il ne faut pas confondre le mode texte et le mode mathématique :

*Exercice 5.* Considérons  $\phi, \Sigma, \hbar, \varepsilon$  et  $\ell$  des réels et  $(O, \vec{i}, \vec{j})$  un repère orthonormé.

---

9. Voir l'annexe B.4.

10. Cette commande ne peut pas mettre en gras tout type de symbole : un package présenté en annexe B.3.3 permet d'y remédier.

Cependant, il arrive que l'on ait besoin d'écrire du texte dans un mode mathématique ; le mieux est alors d'utiliser la commande `\text{}`, qui permet de faire comme si on était en mode texte<sup>11</sup>.

Pour tout $x > 0$ , <code>\[ \Gamma(x)</code> <code>\text{ est bien définie.}</code> <code>\]</code>	Pour tout $x > 0$ , $\Gamma(x)$ est bien définie.
--	--

*Exercice 6* (avec `\overrightarrow{}`). Écrivons le moment magnétique

$$\vec{M} = \frac{1}{2} \iiint_{\mathcal{V}} \vec{OP} \wedge \vec{j}(P) d\tau \quad (\mathcal{V} \text{ étant un volume}).$$

*Exercice 7* (avec `\overset{}`). L'exercice 3 peut aider au calcul de l'intégrale de Fresnel

$$\varphi \stackrel{\text{déf}}{=} \int_0^{+\infty} \exp(ix^2) dx = \frac{\sqrt{\pi}}{2} \exp\left(\frac{i\pi}{4}\right)$$

en montrant, pour  $\alpha$  dans  $]0, 1[$ , que

$$J: \alpha \mapsto \int_{]0, +\infty[} t^{\alpha-1} e^{it} dt$$

vérifie

$$J(\alpha) = \Gamma(\alpha) e^{i\alpha\frac{\pi}{2}}.$$

*Exercice 8* (avec `\xrightarrow{}`). Si  $f \in L^1(\mathbb{R})$  alors sa transformée de Fourier, notée  $\hat{f}$ , est continue et vérifie (pour une définition bien choisie)

$$\hat{f}(x) \xrightarrow{x \rightarrow \pm\infty} 0 \quad \text{et} \quad \left\| \frac{\hat{f}}{2\pi} \right\|_{\infty} \leq \|f\|_1.$$

*Exercice 9* (avec `\ldots`, `\{\displaystyle\}`). Soit  $a_1, \dots, a_k \in \mathbb{N}^*$ . Supposons les  $a_i$  premiers entre eux dans leur ensemble (pour  $i \in \{1, \dots, k\}$ ) et notons, pour  $n \geq 1$ ,  $u_n$  le nombre de  $k$ -uplets  $(x_1, \dots, x_k) \in \mathbb{N}^k$  tels que  $\sum_{i=1}^k a_i x_i = n$ .

Alors

$$u_n \underset{+\infty}{\sim} \frac{1}{a_1 a_2 \cdots a_k} \frac{n^{k-1}}{(k-1)!}.$$

**Conseil.** Ne pas abuser du `\displaystyle` (comme il modifie la dimension de l'interligne, cela peut vite devenir moche!). Prendre garde à la syntaxe : `\{\displaystyle blabla\}` est correct (alors que `\displaystyle\{blabla\}` n'a aucun sens).

11. Certains ouvrages proposent l'utilisation de `\mbox{}` ou de `\mathrm{}` pour cet usage : cela ne produit pas toujours l'effet désiré, donc c'est à proscrire.

### 3.4 Environnements

Voyons maintenant un outil utile pour réaliser des objets plus évolués, comme des tableaux, des équations alignées, des énumérations, etc. : ce sont les environnements. Nous aborderons plus loin (cf. § 3.6) ceux spécifiques au mode mathématique.

Leur utilisation est simple :

```
\begin{<nom de l'environnement>}
:
\end{<nom de l'environnement>}
```

Citons les plus courants :

- `bfseries`, `sffamily`, etc. qui permettent de modifier la mise en forme des fontes, *i.e.* mise en gras, en sans sérif, etc. (cf. § 3.2) ;
- `small`, `Large`, etc. qui permettent de modifier la taille des fontes (cf. § 3.2) ;
- `center`, qui permet de centrer une partie de texte horizontalement ;
- `itemize`, `enumerate` et `description`, qui servent à mettre en forme des listes ;
- `tabular`, qui s'utilise pour créer des tableaux au sein du mode texte (en mode mathématique, c'est son homologue `array` qui doit être utilisé).

Expliquons à l'aide d'exemples le fonctionnement de quelques environnements :

```
\begin{itemize}
  \item premièrement ;
  \item deuxièmement.
\end{itemize}
```

– premièrement ;  
– deuxièmement.

```
\begin{enumerate}
  \item Primo.
  \item Secundo.
\end{enumerate}
```

1. Primo.  
2. Secundo.

```
\begin{tabular}{|r|l|c|}
\hline
right & left & center \\
Xavier & Martine & François \\
\hline
Résistance & Inductance & Capacité \\
\hline
\end{tabular}
```

right	left	center
Xavier	Martine	François
Résistance	Inductance	Capacité

Explications sur la syntaxe de `tabular` : le `\\` indique une fin de ligne, le `&` sépare les colonnes, le `\hline` trace un trait horizontal entre deux lignes. Quant à l'argument de l'environnement (ici `{|r|l|c|}`), il indique deux choses : l'alignement horizontal dans chaque colonne (via la lettre `r`, `l` ou `c`) et l'éventuel ajout d'un filet vertical entre deux colonnes (en insérant le caractère `|`).

*Exercice 10* (on ne demande pas de reproduire l'espacement plus aéré des itemize de cet exercice). Pour avoir la valeur d'une intégrale, deux moyens existent :

1. Calculer sa valeur exacte. Différents outils peuvent être utilisés, en particulier :
  - la règle des invariants de Bioche :
    - si  $-x \leftarrow x$  est un invariant, on utilise  $u = \cos x$ ,
    - si c'est  $\pi - x \leftarrow x$ , on utilise  $u = \sin x$ ,
    - si c'est  $\pi + x \leftarrow x$ , on utilise  $u = \tan x$ ;
  - le théorème des résidus ;
  - l'égalité de Plancherel-Parseval.
2. Calculer une valeur approchée. On distingue deux types de méthodes :
  - (a) des méthodes déterministes, contenant :
    - i. les méthodes de Newton-Cotes,
    - ii. les méthodes de Gauss ;
  - (b) une méthode probabiliste : la méthode de Monte-Carlo.

*Exercice 11.* À savoir sur les méthodes de quadrature :

Méthode	Ordre
Rectangles à gauche	0
Rectangles à droite	0
Point milieu	1
Trapèzes	1
Simpson	3

*Exercice 12* (sans package additionnel mais en utilisant `\multicolumn{}{}{}`). Voici un parallèle entre des méthodes de calcul approché d'intégrale et des schémas de résolution approchée d'équations différentielles ordinaires :

Méthode de quadrature		Schéma EDO
Nom	Ordre	Nom
Rectangles à gauche	0	Euler explicite
Rectangles à droite	0	Euler implicite
Point milieu	1	Euler modifié
Trapèzes	1	Crank-Nicolson
Simpson	3	Runge-Kutta d'ordre 4 (RK4)

Il existe beaucoup d'autres possibilités d'agencer les tableaux : avoir une ligne horizontale ne s'étendant que sur quelques colonnes, modifier la hauteur des cellules, fusionner des cellules, modifier le symbole affiché entre chaque colonne, etc. Enfin, des packages étendent encore ces possibilités... on renvoie au chapitre 4 de [B-Ch].

### 3.5 Sectionnement et références croisées

$\LaTeX$  est capable de gérer tout seul la mise en forme des titres de paragraphes et leur numérotation ; on utilise pour cela les commandes de sectionnement. Il est alors possible de faire des renvois à ces paragraphes : on parle de références croisées.

Pour le sectionnement, les commandes `\chapter{}`, `\section{}`, `\paragraph{}`, etc. (cf. [lshort-fr]) sont à notre disposition ; quant à leur version étoilée (`\section*{}`, etc.), elle supprime la numérotation. L'exemple suivant illustre leur fonctionnement (sous la classe `article`, `\chapter{}` n'est pas disponible).

<code>\section{Première section}</code>	<b>1 Première section</b>
<code>\subsection{Première sous-section}</code>	<b>1.1 Première sous-section</b>
<code>\section{Deuxième section}</code>	<b>2 Deuxième section</b>
<code>\section*{Conclusion}</code>	<b>Conclusion</b>

*Exercice 13.* Recopier l'exemple de code source ci-dessus puis compiler. Ajouter ensuite une section avant la « première section » et compiler de nouveau. Qu'observez-vous ?

Les références croisées permettent de référencer automatiquement les commandes de sectionnement : on place une étiquette juste après le titre (au moyen de `\label{}`) et on s'y réfère *n'importe où* dans le document (au moyen de `\ref{}` pour avoir le numéro du paragraphe et de `\pageref{}` pour avoir la page où il débute). Pour écrire cette feuille de TP, nous avons tapé

```
\section{Découverte de \LaTeX}
\label{sec-Decouv}
```

pour coder le titre de section en page 4, d'où l'exemple suivant :

La section~ <code>\ref{sec-Decouv}</code> débute page~ <code>\pageref{sec-Decouv}</code> .		La section 3 débute page 4.
---	--	-----------------------------

**Conseil.** Prendre l'habitude de « classer » ses références à l'aide d'un mot-clé (suivi d'un *tiret*<sup>12</sup>) : `\label{<mot-clé>-truc}`, en remplaçant par exemple `<mot-clé>` par `sec` pour une section, `subsec` pour une sous-section, `eq` pour une équation numérotée<sup>13</sup>, etc.

---

12. Éviter l'usage du deux-points (en particulier si `babel/frenchb` est chargé), même si certains ouvrages le proposent.

13. Comme nous le verrons par la suite, plein d'objets (numérotés) peuvent être référencés : notes de bas de pages, environnements `table` et `figure`, théorèmes, ...

**Conseil.** L'espace insécable (codé avec le caractère ~) permet d'empêcher un retour à la ligne à l'endroit utilisé ; prendre l'habitude d'en faire précéder chaque `\ref{}`.

*Exercice 14.* Reprendre l'exercice 13 et ajouter une étiquette à la première section, en y faisant référence dans la deuxième ; ensuite, compiler une (seule) fois : qu'observez-vous ? Compiler une seconde fois et trouver une explication.

L'intérêt de la gestion automatique des références croisées par L<sup>A</sup>T<sub>E</sub>X est majeur : en effet, si au cours de la rédaction un numéro de paragraphe doit changer (suite à l'ajout d'un sectionnement avant) ou si le numéro de page change (suite à l'ajout de texte avant), alors les commandes `\ref{}` et `\pageref{}` se mettent à jour toutes seules<sup>14</sup>.

## 3.6 Mathématiques 2

*Voyons maintenant comment enrichir le mode mathématique en utilisant la structure d'environnement et les références croisées. Ce paragraphe répondra aux besoins suivants : faire des tableaux en mode mathématique, créer des matrices, numéroter une formule, afficher une succession d'équations ; écrire proprement des énoncés de théorèmes numérotés ; savoir où trouver des informations supplémentaires.*

### 3.6.1 Agencement de formules, tableaux et matrices

1. L'environnement `equation` permet d'écrire *une* ligne en mode hors-texte (comme la syntaxe `\[ \dots \]`) mais, en plus, il la numérote :

Soit la fonction  $f$ , définie  
par `\begin{equation}`  
 $f(x)=2x$ .  
`\end{equation}`

Soit la fonction  $f$ , définie par

$$f(x) = 2x. \quad (1)$$

2. L'environnement `array` permet de réaliser des tableaux au sein du mode mathématique, mais il faut au préalable être entré dans ce mode. Son utilisation est très similaire à l'environnement `tabular` ; comme ce dernier, il ne fait pas débiter une nouvelle ligne : par défaut il est centré verticalement sur la ligne en cours, mais on peut changer ce placement au moyen des options `t` ou `b`.

Soit `\begin{array}{|c|c|}`  
`\hline`  
 $f(x,t)$  &  $g(x,y)$   
`\hline`  
 $t\sqrt[3]{x}$  &  $\Re\text{\texttrm{e}}(x+iy)$   
`\hline`  
`\end{array}` un beau tableau.  
Soit `\begin{array}[b]{|c|c|}`  
`...$ un autre tableau.`

Soit 

$f(x,t)$	$g(x,y)$
$t\sqrt[3]{x}$	$\Re(x+iy)$

 un beau ta-  
bleau.  
Soit 

$f(x,t)$	$g(x,y)$
$t\sqrt[3]{x}$	$\Re(x+iy)$

 un autre ta-  
bleau.

<sup>14</sup>. Après deux compilations quand même.

3. Pour produire une matrice, on remplacera avantageusement `array` par un environnement mieux adapté : `matrix`, `pmatrix`, `bmatrix`, `vmatrix`, etc.

<pre>Soit \$L=\begin{pmatrix} 2 &amp; -1 &amp; \\ -1 &amp; 2 &amp; -1 \\ &amp; -1 &amp; 2 \\ \end{pmatrix}\$ une matrice.</pre>	$\text{Soit } L = \begin{pmatrix} 2 & -1 & \\ -1 & 2 & -1 \\ & -1 & 2 \end{pmatrix} \text{ une matrice.}$
---	---

4. Pour écrire des équations sur plusieurs lignes, là encore on remplacera avantageusement `array` par un environnement mieux adapté ; citons en particulier

- `gather`, pour empiler des équations (elles seront numérotées) :

<pre>On a \begin{gather} \Delta V=-\vec{\nabla} \cdot(\nabla\vec{E}), \\ \Delta V+ \frac{\rho}{\varepsilon_0}=0. \end{gather}</pre>	<p>On a</p> $\Delta V = -\vec{\nabla} \cdot (\nabla \vec{E}), \quad (2)$ $\Delta V + \frac{\rho}{\varepsilon_0} = 0. \quad (3)$
---	---

- `align` pour empiler des équations en les alignant verticalement <sup>15</sup> (elles seront numérotées) :

<pre>On a aussi \begin{align} (a+b)^2 &amp;= (a+b)(a+b) \\ &amp;= a^2+2ab+b^2. \end{align}</pre>	<p>On a aussi</p> $(a + b)^2 = (a + b)(a + b) \quad (4)$ $= a^2 + 2ab + b^2. \quad (5)$
--	---

On peut faire plusieurs colonnes d'équations (en rajoutant des `&`, généralement placées avant le symbole de relation ; cf. § 3.6.3) ;

- `multline` pour des formules très longues qui tiennent sur plusieurs lignes ; la première est alignée à gauche, la dernière à droite et celles du milieu sont centrées :

<pre>On a \begin{multline} 55= 1+2+3 \\ +4+5+6+7 \\ +8+9+10. \end{multline}</pre>	<p>On a</p> $55 = 1 + 2 + 3$ $+ 4 + 5 + 6 + 7$ $+ 8 + 9 + 10. \quad (6)$
---	--

Ces trois environnements (ainsi que `equation`) admettent une forme étoilée qui supprime la numérotation :

---

<sup>15</sup>. Proscrire l'usage de `eqnarray`.

```
On a enfin \begin{align*}
(a+b)^2 &= (a+b)(a+b)\\
&= a^2+ab+ba+b^2\\
&= a^2+2ab+b^2 \quad \text{car}\dots
\end{align*}
```

```
On a enfin
(a + b)^2 = (a + b)(a + b)
           = a^2 + ab + ba + b^2
           = a^2 + 2ab + b^2      car...
```

5. L'environnement `split`, un peu à part : il permet, alors que l'on est *déjà* en mode mathématique hors-texte, de scinder une équation (qui est trop longue pour tenir sur une seule ligne) ; il nécessite donc d'être placé dans un environnement mathématique et il n'apporte pas de numérotation. Donnons deux exemples d'utilisation.

(a) On souhaite un alignement vertical (ce que ne procure pas `multline`) :

```
On a encore \begin{equation}
\begin{split}
55 &= 1+2+3+4+5\\
&\quad +6+7+8+9+10.
\end{split}
\end{equation}
```

```
On a encore
55 = 1 + 2 + 3 + 4 + 5
    + 6 + 7 + 8 + 9 + 10.      (7)
```

(b) On est déjà dans un environnement mathématique hors-texte (ce qui empêche l'usage de `multline`) :

```
Les équations d'Euler sont
\begin{gather}
\frac{\partial n}{\partial t} + \nabla_x \cdot (nu) = 0, \\
\begin{split}
\frac{\partial (nu)}{\partial t} + \nabla_x \cdot (nu \otimes u) \\
&\quad + \nabla p = 0, \\
\end{split} \\
\frac{\partial E}{\partial t} + \nabla_x \cdot ((E + p)u) = 0.
\end{gather}
```

```
Les équations d'Euler sont
\frac{\partial n}{\partial t} + \nabla_x \cdot (nu) = 0,      (8)
\frac{\partial (nu)}{\partial t} + \nabla_x \cdot (nu \otimes u) \\
+ \nabla p = 0,      (9)
\frac{\partial E}{\partial t} + \nabla_x \cdot ((E + p)u) = 0.      (10)
```

Pleins d'autres outils existent, par exemple :

- l'environnement `cases` permet des « constructions par cas » (cf. exercice 18) ;
- la commande `\intertext{}` permet d'insérer une ligne de texte entre deux lignes de align *sans* briser l'alignement vertical (cf. § 3.6.3) ;
- la commande `\notag` permet de supprimer la numérotation sur une ligne particulière (cf. § 3.6.3).

*Exercice 15* (avec `\eqref{}`). On a l'identité remarquable :

$$\forall a, b \in \mathbb{N}, \quad (a + b)^2 = a^2 + 2ab + b^2. \quad (11)$$

Il s'agit de l'équation (11).

*Exercice 16* (avec `\vdots`). Pour tout  $(a_1, \dots, a_n) \in \mathbb{K}^n$ , le déterminant de Vandermonde est

$$V(a_1, \dots, a_n) = \begin{vmatrix} 1 & a_1 & \cdots & a_1^{n-1} \\ 1 & a_2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_n & \cdots & a_n^{n-1} \end{vmatrix} = \prod_{1 \leq i < j \leq n} (a_j - a_i).$$

*Exercice 17* (avec `\underbrace{ }{ }`). Soit  $(u_n)_{n \in \mathbb{N}}$  définie par  $u_0 \in ]0, \frac{\pi}{2}]$  et  $\forall n \in \mathbb{N}, u_{n+1} = \sin(u_n)$ . Alors on peut montrer successivement que :

$$\begin{aligned} \lim_{n \rightarrow +\infty} u_n &= 0, \\ u_n &\underset{+\infty}{\sim} \sqrt{\frac{3}{n}}, \\ u_n &\underset{+\infty}{=} \sqrt{\frac{3}{n}} - \underbrace{\frac{3\sqrt{3}}{10} \frac{\ln n}{n\sqrt{n}} + o\left(\frac{\ln n}{n\sqrt{n}}\right)}_{=O\left(\frac{\ln n}{n\sqrt{n}}\right)}. \end{aligned}$$

*Exercice 18* (avec cases). Soit  $f: \mathbb{R} \setminus \{0\} \rightarrow \mathbb{R}$ . On peut prolonger  $f$  par continuité en  $\text{sin}_c$  définie par  $\text{sin}_c(x) = \begin{cases} f(x) & \text{si } x \in ]-\infty, 0[ \cup ]0, +\infty[ \\ 1 & \text{sinon} \end{cases}$ .

### 3.6.2 Théorèmes et énoncés similaires

*L<sup>A</sup>T<sub>E</sub>X* permet de mettre en forme des théorèmes, propositions, remarques, etc., de les numéroter automatiquement et d'y faire référence très simplement, tout cela grâce aux environnements de type `theorem`.

Avant toute chose, chargeons le package `amsthm`<sup>16</sup> ; ensuite, cela se fait en deux temps :

1. Définition et configuration d'un type d'énoncé dans le préambule (c'est ici que l'on choisit divers paramètres : quel nom portera-t-il ? Sera-t-il numéroté à la suite ou en parallèle d'un autre type d'énoncé ? Sera-t-il numéroté par section ?).

<sup>16</sup> Les capacités de *L<sup>A</sup>T<sub>E</sub>X* sont très limitées sans ce package ; à l'inverse, pour davantage de possibilités, utiliser le package `ntheorem` (cf. § 3.6.3).

2. Utilisation au cours du document.

Ainsi, en ajoutant ces deux lignes dans le préambule,

```
\newtheorem{prop}{Proposition}
\newtheorem{cor}[prop]{Corollaire}
```

on obtient les deux exemples suivants.

```
\begin{prop}
  Pour tout  $\alpha \in \mathbb{R}$ ,
  la série  $\sum_{n \geq 1} \frac{1}{n^\alpha}$ 
  converge
  si et seulement si  $\alpha > 1$ .
\end{prop}
```

**Proposition 1.** Pour tout  $\alpha \in \mathbb{R}$ , la série

$$\sum_{n \geq 1} \frac{1}{n^\alpha}$$

converge si et seulement si  $\alpha > 1$ .

```
\begin{cor}[Séries de Bertrand]
\label{cor-Bertrand}
  Pour tout  $(\alpha, \beta) \in \mathbb{R}^2$ ,  $\sum_{n \geq 2} \frac{1}{n^\alpha (\ln n)^\beta}$ 
  converge si et seulement si
   $(\alpha > 1)$  ou  $(\alpha = 1$ 
  \text{ et }  $\beta > 1)$ .
\end{cor}
```

**Corollaire 2** (Séries de Bertrand). Pour tout  $(\alpha, \beta) \in \mathbb{R}^2$ ,

$$\sum_{n \geq 2} \frac{1}{n^\alpha (\ln n)^\beta}$$

converge si et seulement si  $(\alpha > 1)$  ou  $(\alpha = 1$  et  $\beta > 1)$ .

(Dans ce dernier exemple un titre a été donné à l'énoncé, via des crochets ; un `\label{}`, lui aussi facultatif, est utilisé pour pouvoir faire référence au corollaire plus tard.)

*Exercice 19.* Retirer le `[prop]` dans la définition de `cor` et observer que la numérotation (des corollaires) a changé : ils sont dorénavant numérotés indépendamment des propositions.

Pour rédiger une preuve, l'environnement `proof` peut être utilisé :

```
\begin{proof}
  Étude de cas et comparaison
  série/intégrale pour  $\alpha > 1$ .
\end{proof}
```

*Démonstration.* Étude de cas et comparaison série/intégrale pour  $\alpha > 1$ . □

```
\begin{proof}[Preuve du
  corollaire-\ref{cor-Bertrand}]
\end{proof}
```

*Preuve du corollaire 2.* □

Pour varier l'affichage des énoncés (*i.e.* avoir par exemple une mise en forme pour les définitions qui soit différente de celle des théorèmes, propositions, lemmes, ...), on dispose de trois styles de composition : `plain`, `definition` et `remark` ; il est aussi possible de ne pas numéroter un type d'énoncé. Voir l'exercice 20 ci-après pour la syntaxe.

*Exercice 20.* Charger le package `amsthm` (au moyen de `\usepackage{amsthm}` placée juste après `\usepackage[frenchb]{babel}`) et écrire les quelques lignes suivantes dans le préambule.

```

\theoremstyle{plain}
\newtheorem{thm}{Théorème}

\theoremstyle{definition}
\newtheorem{defi}[thm]{Définition}

\theoremstyle{remark}
\newtheorem{exe}{Exemple}
\newtheorem*{rmq}{Remarque}

```

Ensuite, obtenir le résultat présenté dans la figure 2 (ci-dessous), largement emprunté de [Obj]. Que se passe-t-il si la 2<sup>e</sup> ligne de code ci-dessus est remplacée par `\newtheorem{thm}{Théorème}[section]` ?

**Définition 1** (Base hilbertienne). Soit  $(H, \langle \cdot, \cdot \rangle)$  un espace de Hilbert. On dit qu'une famille  $(e_i)_{i \in I}$  est une *base hilbertienne* de  $H$  si elle est orthogonale, normée et totale.

**Définition 2.** Un espace métrique  $(E, d)$  est dit *séparable* si tout ouvert non vide de  $E$  contient au moins un point d'une partie dénombrable de  $E$ .

*Exemple 1.*  $\mathbb{R}^n$  est séparable.

**Théorème 3** (Caractérisation). Soit  $(H, \langle \cdot, \cdot \rangle)$  un espace de Hilbert séparable et  $(e_n)_{n \in \mathbb{N}}$  une famille orthonormée de  $H$ . Les propriétés suivantes sont équivalentes :

1. La famille orthonormée  $(e_n)_n$  est une base hilbertienne.
2. Pour tout  $x \in H$ ,

$$x = \sum_{n=0}^{+\infty} \langle x, e_n \rangle e_n \quad (\text{égalité de Parseval}).$$

3. On a

$$(e_n, \quad n \in \mathbb{N})^\perp = \{0\}.$$

*Remarque.* Le théorème 3 s'étend au cas des espaces de Hilbert non séparables.

FIGURE 2 – Texte à reproduire (sans l'encadrement) dans l'exercice 20.

**Conseil.** Pour faire référence à un énoncé numéroté, *toujours* utiliser `\label{}` et `\ref{}`; ne jamais faire une numérotation « en dure »<sup>17</sup>.

<sup>17</sup>. Autrement dit, *ne jamais écrire* « le corollaire 2 » dans son fichier source.

### 3.6.3 Pour aller plus loin : vers l’infini et au delà !

Donnons ici quelques références commentées apportant des informations pour aller plus loin en mathématiques avec  $\LaTeX$ .

Tout d’abord, ces informations pourraient être réparties en trois couches :  $\LaTeX$  sans packages,  $\LaTeX$  avec les packages de l’ $\mathcal{AMS}$  et  $\LaTeX$  avec d’autres packages ; dans ce TP nous confondons les deux premières, car nous estimons que `amsmath` et `amssymb` sont un minimum nécessaire pour faire des mathématiques. Voici donc quelques idées.

1. Consulter une liste de symboles. Les plus usuels figurent dans [lshort-fr] distribué en cours, mais aussi dans [B-Ch], [LPI], ... Pour une liste exhaustive, voir [Pak] (§ 3, 4 pour les mathématiques).
2. Consulter [amsl doc], le *User’s Guide* du package `amsmath`, pour exploiter à fond ses possibilités. Citons en particulier : § 3 (dont la table 3.1) pour d’autres environnements sur l’agencement des formules, § 4, 5 pour diverses notations mathématiques (coefficients binomiaux, fractions continuées, parenthésages, modulo, ...), § 8 pour des diagrammes commutatifs *simples* (*i.e.* sans diagonale<sup>18</sup>).
3. Utiliser le package `mathtools`, extension du package `amsmath` ( $e_n \xrightarrow[n \rightarrow \infty]{} 0$ ).
4. Consulter [Mathmode], qui est une tentative de rassembler toutes les possibilités (commandes, options, packages, etc.) pertinentes pour faire des mathématiques. Il s’agit d’un document très riche mais de haut niveau ; en particulier, savoir que les parties I (traitant de  $\LaTeX$  sans `amsmath`) et III (traitant de  $\TeX$ ) dépassent largement le cadre de ce TP d’initiation.

Voilà pour les généralités ; maintenant, voici une solution pour étendre les possibilités de `amsthm`, au cas où ce dernier se révélerait insuffisant. Plusieurs autres solutions existent, mais c’est celle qui semble la plus aboutie à l’heure actuelle (mars 2010).

5. Utiliser le package `ntheorem`. La personnalisation des énoncés y est plus facile : passage à la ligne après l’affichage du titre numéroté, affichage d’une liste des théorèmes, ... ; une présentation figure dans [B-Ch] et [LPI]<sup>19</sup>.

**Gros exemple** Enfin, terminons en donnant un gros exemple regroupant plein de choses (utiles, à faire ou à éviter). La présentation du code source est obtenue au moyen du package `listings` ; regardez le résultat de la compilation (situé page 21) et revenez ensuite consulter le source ci-dessous pour savoir ce qui s’y cache<sup>20</sup>.

Lignes à ajouter dans le préambule :

```
1 \usepackage{dsfont} %pour un alphabet different de \mathbb{}
2 \usepackage{esvect} %pour les fleches \vv{} plus jolies que \overrightarrow{}
3
```

---

18. Pour des diagrammes plus compliqués, mais au détriment d’une manipulation moins facile, voir le paragraphe 4.4 de ce TP.

19. Attention toutefois : lire le paragraphe 3 de la documentation du package lorsque `amsmath`, `babel` ou `hyperref` sont aussi utilisés.

20. Pour toute explication, consulter [lshort-fr], [amsl doc], etc. ou votre moteur de recherche favori.



```

x&\neq 0
54 \end{aligned}\right.
\end{equation*}
56
\begin{align}
58 \operatorname{div}\{E\} &= \frac{\rho}{\epsilon_0} & \operatorname{div}\{B\} &= 0 \\
\operatorname{rot}\{E\} &= -\operatorname{partial}_t B & \operatorname{rot} B &= \mu_0 \operatorname{biggl}[\operatorname{vec}\{j\} + \operatorname{eps}_0 \operatorname{partial}_t E \operatorname{biggr}]
60 \end{align}

```

Résultat de la compilation (après ajout de traits horizontaux pour séparer les exemples) :

Ne pas confondre `|`, `\lvert`, `\rvert` et `\mid`; ni `|`, `\lVert` et `\rVert`; ni `\backslash` et `\setminus` :

pas bien	→	bien
$a b$	→	$a \mid b$
$\mathbb{P}(A B)$	→	$\mathbb{P}(A \mid B)$
$\{x \in A   x \neq 0\}$	→	$\{x \in A \mid x \neq 0\}$
$ \lambda x  =  \lambda  \ x\ $	→	$\ \lambda x\  =  \lambda  \ x\ $
$\mathbb{N}^* = \mathbb{N} \setminus \{0\}$	→	$\mathbb{N}^* = \mathbb{N} \setminus \{0\}$

Regardez les deux codes sources distincts pour obtenir ces deux morceaux identiques :

$$\left\{x \in A \mid \frac{x}{10} \ll 1\right\} \quad \left\{x \in A \mid \frac{x}{10} \ll 1\right\}$$

et adoptez l'utilisation des commandes personnelles !

Équations de transport sous formes non conservative et conservative ( $a: \mathbb{R}^n \rightarrow \mathbb{R}^n$ ) :

$$\frac{\partial u}{\partial t} + a \cdot \overrightarrow{\operatorname{grad}} u = 0, \tag{12a}$$

$$\frac{\partial u}{\partial t} + \operatorname{div}(au) = 0. \tag{12b}$$

Nous effectuons les calculs suivants

$$A = \frac{\partial f}{\partial t}(x, v, t) + v \cdot \nabla_x f(x, v, t) + \nabla_v \cdot (F(x, v)f(x, v, t)) \tag{13}$$

$$= \dots$$

$$A = \frac{\partial f}{\partial t}(x, v, t) + v \cdot \nabla_x f(x, v, t) + \nabla_v \cdot (F_I(x, v)f(x, v, t)) + \nabla_v \cdot (F_F(x, v)f(x, v, t))$$

qui donnent, plus lisiblement

$$A = \frac{\partial f}{\partial t} + v \cdot \nabla_x f + \nabla_v \cdot (F_F f) + \nabla_v \cdot (F_I f). \tag{♣}$$

$$x \in \emptyset \iff x \in \emptyset \iff \begin{cases} x \not\leq 0 & \text{et } x \not\geq 0 \\ x \neq 0 \end{cases}$$

$$\operatorname{div} E = \frac{\rho}{\epsilon_0} \qquad \operatorname{div} B = 0 \tag{14}$$

$$\overrightarrow{\operatorname{curl}} E = -\partial_t B \qquad \overrightarrow{\operatorname{curl}} B = \mu_0 \left[ \vec{j} + \epsilon_0 \partial_t E \right] \tag{15}$$

## 4 Fonctionnalités plus avancées

### 4.1 Mise en page et structure du document

Nous présentons ici quelques commandes et fonctionnalités de  $\LaTeX$  permettant d'améliorer la présentation d'un document (se reporter à la bibliographie pour davantage d'informations).

**Notes de bas de page :** elles s'utilisent avec `\footnote{}` comme ceci<sup>21</sup>. Observez que dans un tableau cette commande ne fonctionne pas et trouvez comment faire<sup>22</sup>.

**Titre d'un document et page de garde :** il est possible de formaliser l'affichage du titre d'un document ; cela se fait en deux temps :

1. Donner les informations à  $\LaTeX$  (intitulé, auteur(s), date), de préférence dans le préambule.
2. Demander à  $\LaTeX$  d'afficher le titre : il suffit pour cela d'insérer la commande `\maketitle` à l'endroit voulu (généralement au début du document!).

Par exemple, en ayant écrit dans le préambule

```
%Intitule
\title{TP d'initiation à \LaTeX}

%Auteurs
\author{Prénom \bsc{Nom}\thanks{\texttt{prenom.nom@ens-cachan.org}}}

%Date
\date{22 mars 2010}
```

la commande `\maketitle` produit un titre proche de celui de cette feuille de TP. Observez deux autres moyens de gérer la date en supprimant la dernière ligne ci-dessus puis en la remplaçant par `\date{}`.

Il existe aussi l'environnement `titlepage` qui dédie une page entière au titre.

**Table des matières :** la commande `\tableofcontents` permet d'afficher la table des matières à l'endroit où elle est appelée.

*Exercice 21.* En reprenant l'exercice 13, insérer une table des matières au tout début. Compiler, observer ; compiler et observer de nouveau. Ensuite, faire chacune des étapes suivantes, en observant à chaque fois le résultat de la (des) compilation(s) :

1. Modifier un titre de section et ajouter une section.
2. Ajouter `\section[Titre court]{Titre long, vraiment long}`.
3. Placer `\addcontentsline{toc}{section}{\protect Conclusion}` juste après `\section*{Conclusion}`<sup>23</sup>.

21. Je suis une note de bas de page ; je commence par une majuscule et me termine par un point.

22. Solution : grâce à `\footnotemark` et `\footnotetext{}`...

23. Explication : cette nouvelle commande permet d'insérer une ligne dans la `toc` (*table of contents*), contenant un titre de niveau `section` qui s'appelle `Conclusion`. `\protect` permet de ... protéger ☺.



est une unité relative (1 em = largeur de la lettre M majuscule dans la police courante). On comprend très vite l'immense intérêt de cette unité relative : si l'on est amené à changer la taille des fontes d'un document, ces espaces relatifs s'adapteront automatiquement !

Pour faire des espaces horizontales, on dispose en particulier de : `\`, (espace fine), `\quad` (espace inter-mot classique), `\quad` (cadratin, qui vaut 1 em) et `\qquad` (double cadratin), qui sont des espaces de longueurs prédéfinies ; citons enfin `\hspace{}`, pour avoir une espace de taille quelconque, qui prend en argument une valeur positive ou négative, avec une unité de longueur. En mode mathématique, on dispose aussi de la commande<sup>24</sup> `\!` qui permet de réaliser une espace fine négative.

Pour faire des espacements verticaux, on peut utiliser `\vspace{}` ; mais si l'on souhaite espacer ses paragraphes, on privilégiera `\smallskip`, `\medskip` et `\bigskip`, qui sont des blancs relatifs. Cependant, pour utiliser correctement ces trois dernières possibilités, il faut les appeler en dehors d'un paragraphe, c'est-à-dire avoir une ligne vide avant et après dans le fichier source :

<code>\dots fin de paragraphe.</code>	... fin de paragraphe.
<code>\smallskip</code>	Début de paragraphe...
Début de paragraphe <code>\dots</code>	

Enfin, pour changer de page, on utilisera `\clearpage` (qui a l'avantage de placer tous les flottants<sup>25</sup> en attente, s'il y en a).

## 4.2 Inclusion d'images

*Dans de nombreux documents, on est amené à insérer des images au fil du texte : graphiques, schémas, résultats numériques, etc. Insistons sur le fait qu'on traite ici de l'insertion d'images existantes, et non de leur production<sup>26</sup>.*

Le package le plus utilisé pour insérer des images est `graphicx`<sup>27</sup> : il fournit la commande

$$\text{\includegraphics[<options>]{fichier}}$$

qui permet d'insérer une image existante et d'ajuster l'échelle, l'orientation, etc. Les exemples suivant présentent quelques-unes des options ; on renvoie à [lshort-fr] pour une description plus détaillée. Remarquons que dans ces exemples, l'extension du fichier `img` n'est pas précisée : c'est tout à fait licite, et même préférable (cf. conseil ci-dessous).

Attention, les formats d'image autorisés dépendent du mode de compilation (cf. § B.1.2) :

---

24. En fait, grâce au package `amsmath`, cette commande fonctionne en mode texte.

25. Voir le paragraphe 4.3.

26. Si les graphiques ou autres courbes proviennent naturellement de logiciels scientifiques (`scilab`, `matlab`, `maple`, etc.), la création de dessins figuratifs est plus délicate et fera l'objet du paragraphe 4.4.

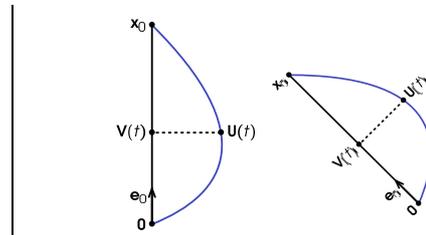
27. Éviter `graphics`, plus ancien.

- compilation via DVI et PS : seules les images au format PostScript sont autorisées (extensions PS, EPS ou EPSF) ;
- compilation avec PDF $\LaTeX$  : les images doivent être dans l'un des formats PNG, PDF ou JPG.

```

\begin{center}
\includegraphics[scale=.4]{img}
\includegraphics[width=2cm,
                 angle=45]{img}
\end{center}

```



Insistons lourdement : une fois qu’une image est insérée, il y a incompatibilité entre ces deux modes de compilation, *i.e.* si une image est insérée au format EPS, alors une compilation avec PDF $\LaTeX$  ne fonctionnera plus.

**Conseil.** Pour contourner cette limitation, nous vous suggérons d’avoir systématiquement chacune de vos images en deux formats (p. ex. EPS et PDF)<sup>28</sup> et de ne pas indiquer d’extension dans le `\includegraphics{}` ; ainsi, selon le mode de compilation effectué,  $\LaTeX$  choisira tout seul l’extension qu’il lui faut pour ne pas rencontrer d’erreurs !<sup>29</sup>

Les images insérées dans  $\LaTeX$  peuvent être numérotées et légendées, voir le paragraphe suivant sur les *flottants*.

### 4.3 Flottants

*Les flottants sont un outil efficace pour placer (dans le document final) des objets déjà créés<sup>30</sup> ; cependant, l’utilisateur se retrouve souvent confronté à des problèmes. Nous présentons ici brièvement ce que sont les flottants et à quel moment des difficultés peuvent intervenir ; ensuite, nous donnons la syntaxe pour créer de tels objets et pour aborder la résolution des problèmes qui s’y rapportent.*

#### 4.3.1 Présentation

Comme nous venons de le voir,  $\LaTeX$  permet de créer des tableaux et d’insérer des images. Ce sont parfois des objets occupant beaucoup de place verticalement et qui peuvent donc poser des problèmes de mise en page. Imaginons par exemple qu’il reste moins de 5 cm de place en bas d’une page au moment où nous souhaitons insérer une figure : que va-t-il se passer si cette dernière occupe plus de 5 cm verticalement ? Nous aimerions la mettre sur la page suivante, où elle aura de la place ; mais alors, un problème apparaît : si nous devons poursuivre la rédaction du document, cela va laisser 5 cm de vide sur la page précédente (ce qui ne donnera pas un rendu très agréable). . .

28. Des logiciels gratuits (tels que The Gimp ou Inkscape) permettent de faire ces conversions.

29. Cela prend bien sûr plus de place sur le disque dur mais évite bien des déboires. . .

30. Tels que des images (cf. § 4.2) ou des tableaux par exemple.

L<sup>A</sup>T<sub>E</sub>X peut remédier à ce problème car il offre la possibilité de faire de cette figure un flottant, c'est-à-dire que sa position dans le document pourra être différente de celle dans le fichier source ; en quelque sorte, on laisse une certaine liberté de placement : on laisse sa figure « flotter ». Dans notre exemple ci-dessus, L<sup>A</sup>T<sub>E</sub>X va débiter une nouvelle page pour insérer la figure mais continuera d'écrire du texte sur la page d'avant, tant qu'il y aura de la place.

Arrive alors le problème fréquent suivant : l'utilisateur n'est pas content des choix de placement faits (par défaut) par L<sup>A</sup>T<sub>E</sub>X ; en effet, un flottant est souvent affiché en début (ou fin) de page, ce qui ne semble pas toujours cohérent. Deux réponses au moins à cela :

- on peut préciser à L<sup>A</sup>T<sub>E</sub>X ses préférences, afin d'influencer ces choix ;
- il faut garder à l'esprit<sup>31</sup> que L<sup>A</sup>T<sub>E</sub>X sait très bien gérer la mise en page, afin que le document soit agréable *à lire* : cette gestion ne satisfait pas toujours l'auteur du premier coup, mais les décisions de L<sup>A</sup>T<sub>E</sub>X sont généralement très performantes<sup>32</sup>.

### 4.3.2 Réalisation

Deux environnements existent pour créer des flottants : `table` et `figure` ; ils font tous les deux sensiblement la même chose, mais le premier est habituellement utilisé pour des tableaux et le second pour des figures, schémas, images, etc. On peut alors

1. Émettre des préférences de placement, parmi les quatre suivantes : ici, en haut de page, en bas de page, sur une page flottante réservée à des flottants ; ces choix se font respectivement via les quatre lettres : `h`, `t`, `b` et `p`. Plusieurs choix peuvent être donnés, par exemple `\begin{table}[htbp]` : L<sup>A</sup>T<sub>E</sub>X essayera de placer le flottant « ici » puis, s'il ne le peut pas, passera aux autres options.
2. Légender un flottant, à l'aide de `\caption{}`.
3. Référencer un flottant, à l'aide de `\label{}` (prendre garde à bien mettre cette étiquette *après* la légende).

Voici un exemple d'utilisation :

```
\begin{figure}[ht]
  \centering
  \includegraphics{image}
  \caption{Position des roues des
trains d'atterrissage principaux sur le Boeing 727}
  \label{fig-Boeing}
\end{figure}
```

---

31. Ou se convaincre...

32. Les règles de placement ainsi que les paramètres pris en compte pour la gestion des flottants sont nombreux et techniques ; cela est donc à double tranchant : le résultat est satisfaisant dans de nombreux cas mais en contrepartie, lorsqu'il ne l'est pas, il devient compliqué de savoir d'où vient le désaccord avec les choix de L<sup>A</sup>T<sub>E</sub>X.

Attention, il faut bien comprendre que `figure` ne sert pas à insérer une image, mais à faire flotter une image (insérée par `\includegraphics{}`).

Plein de contraintes rentrent en compte pour le placement des flottants (nombre maximal de flottants par page, etc.) ; cela fait que les problèmes apparaissent essentiellement dans deux cas : si les flottants sont très gros ou s'il y en a beaucoup<sup>33</sup>. Le premier réflexe peut être de se demander s'il est pertinent de mettre plein de flottants dans son document ; ensuite, on peut agir sur les options de placement du flottant<sup>34</sup>. Enfin, il est aussi possible (mais déconseillé) de relâcher certaines de ces contraintes, via un « ! » : on écrira `[!h]` si l'on souhaite *vraiment* que le flottant soit ici, au détriment des règles esthétiques de L<sup>A</sup>T<sub>E</sub>X.

Lire le paragraphe 2.12 de [short-fr] pour mieux comprendre comment fonctionnent les flottants et pourquoi vous avez un problème. Voir [Loz] pour des recommandations sur l'utilisation des flottants ; une des principales règles étant : ne pas écrire « voir la figure ci-dessous » mais « voir la figure 3 » (en ayant préalablement référencé cette figure).

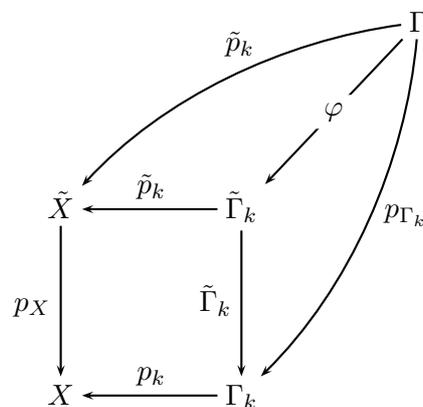
#### 4.4 Schémas et graphiques

Il existe différents outils pour créer des dessins directement dans L<sup>A</sup>T<sub>E</sub>X. À l'heure actuelle, le plus complet est *PSTricks* ; il existe aussi *PGF/TikZ*, présenté dans [short-fr]. Les quelques exemples ci-dessous montrent un aperçu du premier.

Pour l'utiliser, il faut charger le package `pstricks` et parfois des packages auxiliaires. Attention : comme pour l'insertion d'images au paragraphe 4.2, le choix mode de compilation est important ici (cf. § B.1.2) ; *PSTricks* ne fonctionne pas avec PDFL<sup>A</sup>T<sub>E</sub>X.

Avec l'extension `pst-node`, on peut créer des graphes ou diagrammes :

```
\begin{psmatrix}[colsep=2cm,
                 rowsep=2cm]
& & \Gamma \\
\tilde{X} & \{\displaystyle \tilde{\Gamma}_k\} & \\
X & \{\displaystyle \Gamma_k\} & \\
\psset{arrows=->,labelsep=3pt,
       ,nodesep=3pt}
\ncarc[arcangle=-20]{1,3}{2,1}
      ~{\tilde{p}_k}
\nclice{1,3}{2,2}\ncput*{\varphi}
\ncarc[arcangle=20]{1,3}{3,2}
      >{p_{\Gamma_k}}
\nclice{2,2}{2,1}~{\tilde{p}_k}
\nclice{2,1}{3,1}<{p_X}
\nclice{2,2}{3,2}<{\tilde{\Gamma}_k}
\nclice{3,2}{3,1}~{p_k}
\end{psmatrix}
```

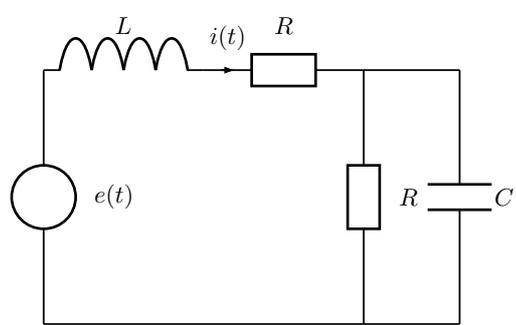


33. En effet, tant qu'un flottant n'est pas placé, tous les suivants (même s'ils ne gênent pas L<sup>A</sup>T<sub>E</sub>X) sont mis en attente.

34. En particulier, s'il y a beaucoup de gros flottants, leur attribuer une page entière (option `p`) peut être utile.

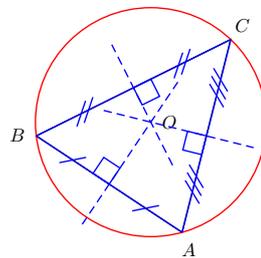
L'extension `pst-circ` fournit des macros pour dessiner des circuits électriques :

```
\begin{pspicture}(7,4.5)
\node(0,0){A}\node(5,0){B}
\node(0,4){C}\node(5,4){D}
\Ucc[labeloffset=-1.1](A)(C){$e(t)$}
\multidipole(C)(D)\coil{$L$}\resistor
[intensitylabel=$i(t)$]{$R$}.
\resistor(D)(B){$R$}
\wire(A)(B)
\capacitor[parallel](D)(B){$C$}
\end{pspicture}
```



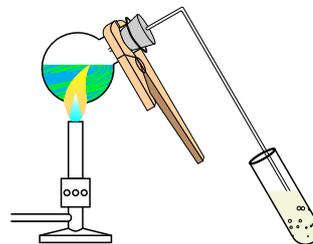
Des figures de la géométrie euclidienne sont prédéfinies dans l'extension `pst-eucl` :

```
\begin{pspicture}(6,6)
\pstTriangle[PointSymbol=none]
(4,1){A}(1,3){B}(5,5){C}
\pstCircleABC[CodeFig=true,
CodeFigColor=blue, linecolor=red,
PointSymbol=none]{A}{B}{C}{O}
\end{pspicture}
```



Citons enfin `pst-labo` pour la chimie :

```
\psset{pince=true,glassType=ballon}
\pstChauffageTube[becBunsen,barbotage,
substance={\pstFilaments[10]{green}}]
```



Il y a aussi la possibilité de réaliser des arbres, des formules et représentations moléculaires, des morceaux de musiques, des figures de diffraction, ...

Mentionnons pour finir qu'il existe des logiciels externes à  $\text{\LaTeX}$  permettant de dessiner à la souris et ensuite de transformer le résultat en un code utilisable par  $\text{\LaTeX}$  :  $\text{\TeXgraph}$ ,  $\text{xfig}$  et plusieurs autres.

## 4.5 Transparents et présentations

$\text{\LaTeX}$  fournit la classe `slides` pour mettre en page des transparents. Toutefois, il existe un meilleur moyen de créer des présentations ou diaporamas au format PDF grâce à

$\LaTeX$  : il s'agit d'un package (qui définit une nouvelle classe  $\LaTeX$ ) nommé `beamer`. La classe `beamer` génère des « transparents » (*frames*) qui peuvent être affichés en plusieurs étapes par une succession de couches (*slides*). Pour obtenir une sortie `beamer`, le début du document est un peu différent de celui habituellement utilisé : la figure 3 en présente un exemple.

```
\documentclass{beamer}

\usepackage[<option à choisir>]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage{amsmath, amssymb}
\usepackage[frenchb]{babel}

\usetheme{Warsaw}
```

FIGURE 3 – Exemple de fichier source `beamer`.

On retrouve les packages liés aux mathématiques ou à la langue française et il existe une ligne supplémentaire qui définit le thème `beamer`. Ici c'est le thème `Warsaw` qui a été choisi (il existe de nombreux autres thèmes dont le détail ici serait fastidieux). Un petit tour sur Internet permet de trouver rapidement le thème voulu. On citera les plus classiques (certains peuvent utiliser des options) :

```
\usetheme{Warsaw}
\usetheme{AnnArbor}
\usetheme[secheader]{Boadilla}
\usetheme[secheader]{Madrid}
\usetheme[compress]{Ilmenau}
```

Après le préambule, on peut définir le titre, le sous-titre, l'auteur, la date, etc. ; viendront ensuite les transparents (*frames*) : un *frame* se compose d'un titre, d'un sous-titre et de texte. Autour de cet ensemble, on trouvera (suivant le thème choisi) des bandes de couleurs, des symboles de navigation, le nom de l'auteur, le titre de la section, un logo, etc. Une image étant toujours plus parlante que des mots, un exemple de *frame* est donné sur la figure 4.

Dans le corps du transparent, on peut intégrer des tableaux, des listes, des formules, des définitions, des théorèmes, des remarques et à peu près tout ce qu'on peut faire classiquement en  $\LaTeX$ . Attention, il existe des exceptions à cette règle, par exemple l'insertion d'un environnement `verbatim` nécessite une option `fragile`. Sans cette option, la compilation échoue.

<pre>\begin{frame}[fragile] Ce texte est normal. \verb_Ici on est en mode verbatim._ \end{frame}</pre>	<pre>Ce texte est normal. Ici on est en mode verbatim.</pre>
--	--



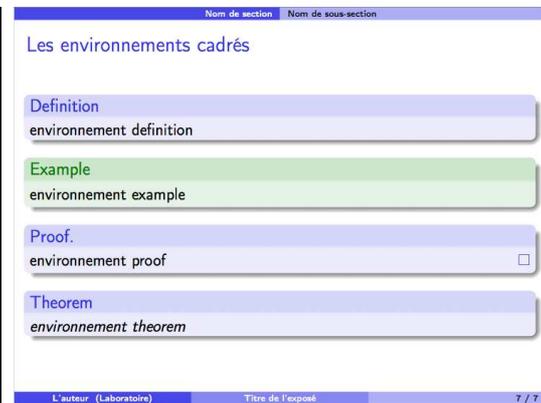
FIGURE 4 – Exemple de frame beamer.

Une utilisation classique de `beamer` est l'emploi des *blocks*. Par défaut, les théorèmes, exemples, définitions et preuves sont dans des blocks.

```

\begin{frame}
  \frametitle{Les environnements
cadrés}
  \begin{definition}
    environnement definition
  \end{definition}
  \begin{example}
    environnement exemple
  \end{example}
  \begin{proof}
    environnement proof
  \end{proof}
  \begin{theorem}
    environnement theorem
  \end{theorem}
\end{frame}

```

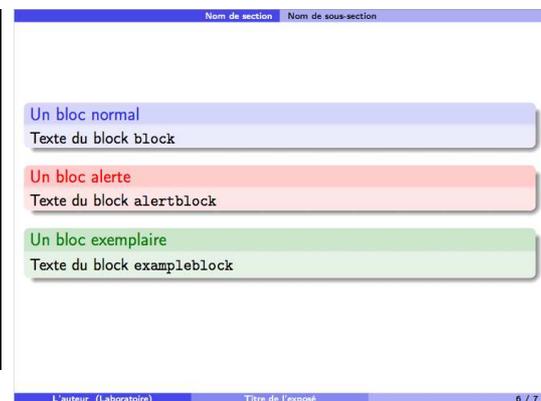


Par ailleurs, il existe d'autres types de blocks, dont les détails peuvent être redéfinis manuellement.

```

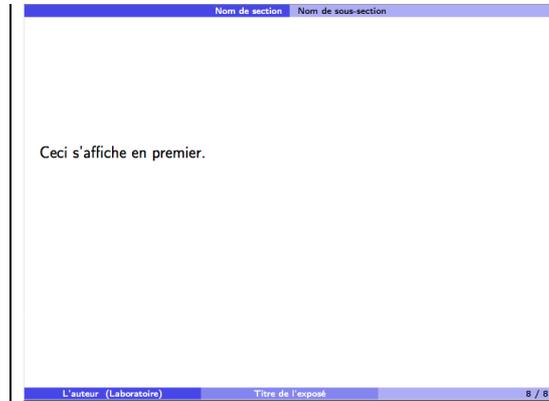
\begin{frame}
  \begin{block}{Un bloc normal}
    Texte du block \texttt{block}
  \end{block}
  \begin{alertblock}{Un bloc alerte}
    Texte du block \texttt{alertblock}
  \end{alertblock}
  \begin{exampleblock}{Un bloc exemple}
    Texte du block \texttt{exampleblock}
  \end{exampleblock}
\end{frame}

```



Enfin, il existe une commande très pratique : la commande `\pause`. Elle permet d'empêcher une partie du transparent d'apparaître : elle gèle en quelque sorte l'affichage complet du transparent.

```
\begin{frame}
  Ceci s'affiche en premier.\\
  \pause
  Cette partie ne s'affiche que
  lorsque je suis pr\^et.\\
  \pause
  Cette derni\`ere partie
  appara\^it encore apr\`es.
\end{frame}
```



On peut se demander à quoi cela sert, car il suffirait de faire 3 transparents différents pour obtenir le même résultat. Passant le fait que cela est beaucoup plus contraignant, on note en particulier que le compteur de transparents ne s'incrémente pas. De plus, de nombreuses options permettent de changer l'ordre d'affichage, revenir en arrière, modifier la forme du texte de manière temporaire, etc. C'est ici qu'apparaît la notion de couches, notion quelque peu complexe qu'on n'abordera pas ici. Pour plus de détails, on consultera l'excellente page [McC].

## A Typographie française

*L'objet de cette annexe est de présenter quelques-unes des règles les plus élémentaires de la typographie française et leur mise en forme (simplifiée) grâce à  $\LaTeX$ .*

### A.1 Avec $\LaTeX$

Le paragraphe 2.5.2 de [lshort-fr] explique les outils fournis par `babel/frenchb` (voir le paragraphe 2 de [Fli] pour des explications complètes) ; nous utilisons quelques-unes des abréviations proposées par `babel/frenchb` en page 33 du TP.

### A.2 Quelques règles et usages

#### A.2.1 Espacement et ponctuation

Grâce à `babel/frenchb`, il suffit d'utiliser la « barre d'espacement » du clavier chaque fois qu'une espace est nécessaire ; en effet,  $\LaTeX$  saura alors la transformer automatiquement en le type d'espace adéquat :

- virgule et point sont suivis d'un blanc ;
- point d'interrogation, point d'exclamation, point-virgule et deux-points sont suivis d'un blanc et précédés d'une « espace fine insécable » (sauf le deux-points, en général précédé d'une « espace insécable ») ;
- les guillemets ouvrants ou fermants sont respectivement, précédés ou suivis d'un blanc, suivis ou précédés d'une espace insécable ;
- les parenthèses ou crochets ouvrants sont précédés d'un blanc ;
- les parenthèses ou crochets fermants sont suivis d'un blanc ;
- les apostrophes et traits d'union ne sont ni précédés ni suivis de blanc ;
- le tiret (codé `---`) utilisé pour ouvrir ou fermer une incise est respectivement, précédé ou suivi d'un blanc, suivi ou précédé d'une espace fine insécable ;
- le point de suspension (codé `\dots`) est suivi d'un blanc (pour que ce blanc apparaisse, on rajoute des accolades après le nom de la commande : `\dots{}`).

#### A.2.2 Abréviations

- Une abréviation qui ne se compose que des premières lettres du mot se termine par un point (référence = réf.).
- Une abréviation qui se termine par la dernière lettre du mot ne comporte pas de point final (boulevard = bd).
- L'abréviation des groupes de mots ne comporte pas de point final (s'il vous plaît = svp).

Voici quelques exceptions souvent utilisées :

c'est-à-dire	c.-à-d. ou c-à-d
<i>confer</i>	cf.
environ	env.
<i>et cætera</i>	etc. (pas de point de suspension ni de répétition)
exemple	ex.
figure	fig.
<i>idem</i>	<i>id.</i>
<i>id est</i>	<i>i. e.</i>
page	p. (ne se répète pas : p. 127 et 128)
paragraphe	§ (ne se répète pas : § 4 et 5)
numéro, numéros	n <sup>o</sup> , n <sup>os</sup> (et N <sup>o</sup> , N <sup>os</sup> )
premier, premiers	1 <sup>er</sup> , 1 <sup>ers</sup>
première, premières	1 <sup>re</sup> , 1 <sup>res</sup>
deuxième, deuxièmes	2 <sup>e</sup> , 2 <sup>es</sup>
primo, secundo, tertio	1 <sup>o</sup> , 2 <sup>o</sup> , 3 <sup>o</sup>
degré (Celcius)	39 °C (espace insécable)
degré (alcoolique)	40° (pas d'espace en français)
monsieur, messieurs	M., MM. (pas de « Mr », abréviation de l'anglais <i>mister</i> )
madame, mademoiselle	M <sup>me</sup> , M <sup>lle</sup>

L'abréviation « § » est fournie par L<sup>A</sup>T<sub>E</sub>X : `\S` ; les autres se codent facilement grâce à `babel/frenchb` : `\no{}`, `1\iers{}`, `2\ieme{}`, `\primo{}`, `39-\degres{}` C, `M\up{me}`, ...

### A.2.3 Ponctuation

Un mauvais emploi de la ponctuation peut rendre difficile la lecture, ou même changer le sens d'une phrase. Quelques rappels :

- les phrases se terminent par un point final, un point d'exclamation, un point d'interrogation ou un point de suspension ;
- la virgule sépare les parties d'une proposition ou d'une phrase à condition qu'elles ne soient pas déjà réunies par les conjonctions *et*, *ou*, *ni*. Elle sépare les énumérations. Elle isole les mots mis en apostrophe. On ne met pas de virgule avant une parenthèse, un tiret ou un crochet ;
- le point virgule sépare différentes propositions de même nature ;
- le deux-points introduit une explication, une citation, un discours, une énumération ;
- le point de suspension est toujours composé de trois points ;
- les parenthèses intercalent une précision dans la phrase ;
- les crochets indiquent une précision à l'intérieur d'une parenthèse ou une coupure dans une citation ;

- employé seul, le tiret signale chaque terme d'une énumération ou le changement d'interlocuteur dans un dialogue. Employés par paires, ils s'utilisent comme les parenthèses ; en fin de phrase, le tiret fermant est supprimé avant le point final.

#### A.2.4 Énumérations

On peut distinguer deux types d'énumérations :

- celles qui arrivent en cours de phrase :
  - elles sont introduites par un deux-points et chaque terme commence par une minuscule,
  - celles de premier rang sont introduites par un tiret et se terminent par un point-virgule (même s'il y a plusieurs phrases dans l'énumération), sauf la dernière par un point final,
  - celles de second rang sont introduites par un tiret décalé et se terminent par une virgule ;
- celles qui débutent une phrase : chaque terme commence alors par une majuscule et se termine par un point (voir un exemple en A.2.5).

En général, lorsque chaque élément de l'énumération est introduit par un numéro (ou autre symbole) suivi d'un point, on applique toujours le 2<sup>e</sup> cas ci-dessus.

#### A.2.5 Divers

- On rappelle que l'accent a pleine valeur orthographique ; en conséquence **on met les accents sur les majuscules**.
- On écrit « on résout », « quel que soit », « quelles que soient », « intéresser » et « on rappelle qu'il est rappelé ».

Enfin, on n'oubliera pas d'être informé de la « nouvelle orthographe », réforme orthographique de l'Académie Française datant de 1990 ; l'emploi de ces rectifications n'est pas imposé, mais il est recommandé (en particulier, aucune des deux graphies ne peut être tenue pour fautive). Citons deux des dix nouvelles règles :

- « On emploie l'accent grave (plutôt que l'accent aigu) dans un certain nombre de mots (pour régulariser leur orthographe), au futur et au conditionnel des verbes qui se conjuguent sur le modèle de céder, et dans les formes du type puissè-je ». Ex : évènement, je céderai ;
- « L'accent circonflexe disparaît sur *i* et *u*. On le maintient néanmoins dans les terminaisons verbales du passé simple, du subjonctif et dans cinq cas d'ambiguïté » (motivation : « sur *i* et *u*, l'accent circonflexe ne joue aucun rôle phonétique ; il est l'une des principales causes d'erreurs et son emploi, aléatoire, ne peut être justifié par l'étymologie. »). Ex. : cout, il paraît ; dû, sûr, mûr, jeûne.

Ces quelques lignes sont issues de [Ortho], court document qui résume cette réforme, l'explique, en décrit les avantages et donne des remarques ainsi que des exemples.

## B Explications complémentaires au TP

*Dans cette annexe, nous expliquons certains points non détaillés pendant le TP, afin de mieux les comprendre et de savoir les utiliser efficacement. Pour des explications plus exhaustives, se reporter aux ouvrages sur  $\LaTeX$ .*

### B.1 Compréhension et utilisation de $\LaTeX$

#### B.1.1 Fonctionnement et installation

*Dans ce paragraphe, nous revenons plus en détail sur les quatre étapes décrites à la section 1, après avoir donné quelques explications sur l'installation de  $\LaTeX$ .*

**Installation sous Windows** Différents outils sont à installer, et dans le bon ordre. Nous recommandons d'utiliser Pro $\TeX$ t (<http://www.tug.org/protext>, `protext.exe`), qui regroupe ces outils en un :

$$\text{Pro}\TeX\text{t} \simeq \text{MiK}\TeX + \text{T}\TeX\text{nicCenter} + \text{Ghostscript} + \text{GSview},$$

ces deux derniers assurant la gestion du format PostScript. L'installation est entièrement guidée par un PDF interactif qu'il suffit de suivre pour installer chacun des composants (nous recommandons vivement de lire attentivement ce court document avant de débiter l'installation) ; il est possible d'utiliser un autre éditeur de texte, comme Texmaker (<http://www.xmlmath.net/texmaker/>).

**Installation sous Mac OS** Nous recommandons d'installer Mac $\TeX$  (<http://www.tug.org/mactex>), qui contient l'interface graphique  $\TeX$ Shop que nous conseillons d'utiliser :

$$\text{Mac}\TeX \simeq \text{T}\TeX\text{Live} + \text{T}\TeX\text{Shop}.$$

**Installation sous Linux** Nous recommandons la distribution  $\TeX$ Live (<http://www.tug.org/texlive>) avec l'éditeur Kile. La procédure d'installation dépend de la distribution Linux (cf. [LPI]).

**Les quatre étapes** L'étape 1 pourrait se faire via un éditeur « quelconque », comme Nedit (sous Linux), TextEdit (sous Mac OS) ou WordPad (sous Windows) ; mentionnons aussi Emacs, éditeur de texte non graphique mais polyvalent et disponible sur tous les systèmes d'exploitation : ses adeptes sont nombreux (la saisie d'un fichier `.tex` est facilitée grâce à l'extension Auc $\TeX$ ). Les autres étapes se feraient alors via une fenêtre de commande (où l'on taperait « à la main » `latex TP.tex` pour compiler). Néanmoins, il existe des éditeurs de texte orientés  $\LaTeX$  qui permettent de faire ces étapes beaucoup plus agréablement, en quelques clics : c'est ce que nous avons utilisé pendant la séance de TP et nous conservons ce point de vue dans la suite, où nous donnons quelques remarques propres à chaque étape.

Pour saisir son document (étape 1), plusieurs interfaces graphiques existent, dont

- sous Linux : Kile, Texmaker ;
- sous Mac OS : T<sub>E</sub>XShop, Texmaker ;
- sous Windows : T<sub>E</sub>XnicCenter, Texmaker, WinEdt<sup>35</sup>.

Ces interfaces graphiques (ou éditeurs de texte) sont l’endroit où l’on tape son fichier source ; ils sont à distinguer de « ce qui fait fonctionner L<sup>A</sup>T<sub>E</sub>X » à proprement parler, c’est-à-dire de ce qui fait fonctionner la compilation (étape 2). On parle alors de *distributions* L<sup>A</sup>T<sub>E</sub>X ; ces dernières ne sont pas non plus les mêmes selon les systèmes d’exploitation. Citons

- sous Linux : T<sub>E</sub>XLive ;
- sous Max OS : T<sub>E</sub>XLive ;
- sous Windows : MiK<sub>T</sub>E<sub>X</sub>.

Après la compilation, L<sup>A</sup>T<sub>E</sub>X a produit le fichier TP.dvi ; il en a aussi créé d’autres (qui lui ont été utiles), mais les éditeurs de texte savent choisir le bon lorsque l’on demande de visualiser (étape 3). Le logiciel de visualisation est en général fourni avec la distribution L<sup>A</sup>T<sub>E</sub>X (Yap pour Windows, xdvi ou kdvi pour Linux, xdvi pour Mac) ; ce format DVI nécessite cependant que L<sup>A</sup>T<sub>E</sub>X soit installé pour s’ouvrir correctement : c’est en quelque sorte un outil de travail pour le rédacteur, un peu comme un brouillon<sup>36</sup>. Par contre, une fois que le document est sous sa version finale, on souhaite en général l’imprimer ou le diffuser (étape 4) : on utilise alors le format PDF (le format PS existe aussi). Les interfaces graphiques citées ci-dessus permettent de faire ces conversions en quelques clics : on obtient alors un fichier lisible et imprimable par tous !

### B.1.2 Compilation et obtention du PDF

Le schéma suivant résume les étapes de réalisation :

$$\text{TEX} \longrightarrow \text{DVI} \longrightarrow \text{PS} \longrightarrow \text{PDF}.$$

Une dernière remarque s’impose alors : il existe un mode de compilation (PDFL<sup>A</sup>T<sub>E</sub>X) qui, comme son nom l’indique, compile directement TP.tex en TP.pdf :

$$\text{TEX} \longrightarrow \text{PDF}.$$

Attention : cet autre type de compilation *n’est pas un simple raccourci* des étapes précédentes. Certains fichiers sources ne se compileront pas correctement avec PDFL<sup>A</sup>T<sub>E</sub>X (notamment si l’on utilise PSTricks<sup>37</sup>, qui a besoin – en général – d’un passage par le format PS), alors que d’autres seront plus faciles à compiler via PDFL<sup>A</sup>T<sub>E</sub>X (comme la classe beamer et plus généralement tout fichier exploitant les fonctionnalités de la norme PDF<sup>38</sup>). Il est surtout très important de faire attention au type de compilation

---

35. Ce dernier est payant.

36. On remarquera que ces logiciels mettent automatiquement à jour le fichier DVI après une nouvelle compilation : il est donc inutile de les redémarrer à chaque fois.

37. Outil le plus complet pour faire des graphiques évolués (voir le paragraphe 4.4).

38. Comme les liens hypertextes, les propriétés d’affichage par défaut du PDF, etc.

lors de l'inclusion d'images ou de graphiques : PDF $\LaTeX$  ne permet d'incorporer que des fichiers JPEG, PNG et PDF, alors que l'autre chemin ne permet d'incorporer que des fichiers PostScript (format PS ou EPS). Beaucoup de logiciels mathématiques proposent d'enregistrer leur image en `.eps` ; de plus, des logiciels graphiques (tels que `Inkscape`, `TheGimp` ou `PhotoFiltre`) permettent de faire des conversions vers une telle extension. Signalons enfin que certains packages existent pour contourner tous ces obstacles<sup>39</sup> ; en particulier, l'option `dvips` du package `hyperref` permet de conserver les liens interactifs d'un document (`beamer` ou autre) lorsque l'on crée le PDF en passant par DVI + PS.

**Que retenir de tout cela ?** Les procédés mis en jeux par  $\LaTeX$  ne sont pas simples ; il est cependant inutile d'en connaître toutes les subtilités pour débiter, car les interfaces graphiques fournissent un environnement complet pour la réalisation de documents. Quant aux deux types de compilation ci-dessus, le plus simple nous paraît de faire un choix et de *s'y tenir dans un maximum de cas* ; ne faire alors appel à l'autre chemin que lorsqu'il semble incontournable.

## B.2 Le `\documentclass`

Le `\documentclass` permet de donner la *classe* (ou le *type*) du document que l'on souhaite rédiger ;  $\LaTeX$  en propose au moins cinq (`article`, `report`, `book`, `letter`, `slides`) mais il en existe d'autres (comme `beamer`). Nous ne présentons ici que les trois premières (voir paragraphe 4.5 pour `slides` et `beamer`) :

- `article` convient pour un document court (moins de 30 pages) tel qu'un article de recherche, un petit rapport<sup>40</sup>, une documentation, etc. ;
- `report` convient pour un document de taille moyenne, tel qu'un long rapport, un mémoire, etc. ;
- `book` est à réserver pour un long document tel qu'un livre, un rapport de thèse, un polycopié de cours, etc.

En fait, le choix de la classe n'est pas aussi rigide : chaque classe offre quelques fonctionnalités différentes (voir ci-dessous) et c'est selon ses préférences que l'on choisit la classe pour les cas limites (comme un rapport de stage d'une trentaine de pages. . .).

Des options peuvent être ajoutées à la déclaration de classe, en particulier :

- la taille du corps de base : `10pt` (par défaut), `11pt` ou `12pt` ;
- l'option `titlepage` qui indique qu'une nouvelle page doit être commencée après l'ajout du titre par `\maketitle`<sup>41</sup> ;
- l'option `draft` qui permet, entre autres, d'éviter de charger les images incluses par `\includegraphics` lors de la pré-visualisation (tout en délimitant dans le document l'espace nécessaire à l'insertion des images) : cela évite les ralentis de com-

---

39. Voir les packages `hyperref`, `urlbreak`, `pdftricks`, . . . ce dernier n'étant pas efficace dans 100% des cas.

40. Rapport de TP, de séminaire, etc.

41. Voir paragraphe 4.1.

pilation et d’affichage, sans modifier la mise en page. Cette option est bien sûr à retirer avant la compilation finale !

Elles s’utilisent suivant la syntaxe

```
\documentclass[12pt, titlepage]{article}.
```

Citons maintenant deux différences entre ces trois classes : tout d’abord, `report` et `book` admettent le sectionnement `\chapter`, contrairement à `article`<sup>42</sup>. Ensuite, L<sup>A</sup>T<sub>E</sub>X choisit les options par défaut et ces choix peuvent dépendre de la classe (ainsi `report` et `book` sont par défaut sous `titlepage`, alors qu’`article` ne l’est pas). Bien souvent, il est déconseillé de modifier un grand nombre d’options car, pour une classe de document donnée, les choix de L<sup>A</sup>T<sub>E</sub>X sont habituellement judicieux du point de vue de la mise en page.

## B.3 Le préambule et les packages

### B.3.1 Les packages « de base »

Nous revenons ici sur les packages présentés à la section 2, en décrivant très brièvement leur rôle ; des informations complètes peuvent être trouvées dans la documentation de chaque package (celle-ci étant disponible sur le Web mais aussi, en général, copiée sur le disque dur à l’installation de L<sup>A</sup>T<sub>E</sub>X).

- `inputenc` permet de taper directement au clavier les accents et cédilles dans le fichier source (ce package sert à spécifier le codage d’entrée). Faire attention : l’option dépend du système d’exploitation ou de l’éditeur de texte sous lequel est *saisi* (ou *a été saisi*) le fichier ; revenons donc sur le choix de l’option d’`inputenc`, comme écrit page 4 :
  - sous Linux, l’encodage par défaut peut être UTF-8 (remplacer alors *<option à choisir>* par `utf8`) ou ISO-8859-1 (remplacer alors par `latin1`) ;
  - sous Mac OS, il s’agit de l’encodage Mac OS Roman (remplacer alors par `applemac`) ;

Pour savoir dans quel encodage est votre éditeur, il faut regarder dans un menu du genre « options » ou « préférences ». Dans tous les cas, il faut que l’option donnée à `inputenc` coïncide avec le réglage de l’éditeur !

Insistons lourdement et remarquons que cela nuit gravement à la portabilité du fichier source (mais pas du PDF) ; exemple : si l’individu *M* tape son fichier source sous Windows et que l’individu *G* souhaite le compiler sur Mac OS, alors il n’y aura pas de problème (à condition que *G* laisse l’option `cp1252`) ; mais si *G* veut modifier le fichier sur son Mac, alors il y aura un mélange des encodages et les accents des fichiers (source et compilé) ne s’afficheront pas correctement. Un moyen efficace (mais pénible) de palier ce problème est de transformer tous les caractères accentués en leur « équivalent

---

42. L<sup>A</sup>T<sub>E</sub>X estime qu’un document de classe `article` n’a pas besoin de ce niveau de sectionnement : si au cours d’une rédaction sous `article` vous en ressentez le besoin, c’est qu’il faut changer de classe.

TeX » (é  $\mapsto$  `\'e`, etc.) au moment de l'échange du fichier source... ; une autre solution, plus simple, consiste à se mettre d'accord sur un encodage commun.

Nous conseillons à l'heure actuelle aux utilisateurs de Linux et de Mac OS d'utiliser le codage ISO-8859-1 dans leur éditeur de texte (choix discutable, cf. § B.3.2).

- `fontenc`, avec l'option `T1`, permet une bonne césure<sup>43</sup> des mots de langues latines (ce package sert à spécifier le codage des fontes).
- `lmodern` permet d'adapter la police de base de L<sup>A</sup>T<sub>E</sub>X aux nouveaux formats apparus depuis sa création, le format PDF notamment (ce package charge la police *Latin Modern*).
- `babel` permet de prendre en compte la langue de rédaction (éventuellement plusieurs) pour la mise en page, les règles typographiques, les césures, etc. ; `frenchb` est l'option actuellement à utiliser pour un document en français. On notera que `babel/frenchb` apporte des commandes supplémentaires pour la langue française (par exemple, `\ier{}` permet d'écrire l'exposant de 1<sup>er</sup>) et francise les mots-clés de L<sup>A</sup>T<sub>E</sub>X (par exemple, l'appel à `\tableofcontents` affichera « Table des matières » au lieu « Contents »).
- `geometry` permet ici de spécifier la dimension des pages du document (l'option par défaut étant le format en vigueur aux États-Unis, il est important de préciser l'option `a4paper` pour une impression qui se fera sur du papier A4). Ce package est bien plus puissant : il facilite aussi un contrôle fin et complet des marges du document (essayez `[a4paper, body={16cm,24cm}]`).
- `amsmath` et `amssymb` apportent des fonctionnalités supplémentaires en mode mathématique. Remarquez la syntaxe (nous aurions pu aussi les charger au moyen de deux `\usepackage{}`) : en regroupant de la sorte, on peut rendre plus lisible le fichier source<sup>44</sup>.

### B.3.2 Package `inputenc` et encodage

Ce paragraphe est l'occasion de parler brièvement d'encodage. C'est une notion technique, déjà abordée en note n° 3 (page 4) et au paragraphe B.3.1 (avec `inputenc`) ; on a besoin de s'en soucier dans trois cas au moins : lorsque plusieurs personnes travaillent sur un même fichier `.tex` via des systèmes d'exploitation différents ou lorsqu'une personne travaille son fichier sur des systèmes d'exploitation différents (par exemple Windows à la maison et Linux à la fac) ou lorsque la langue de rédaction utilise des caractères non présents dans notre alphabet « classique ». À l'heure actuelle (mars 2010) *aucun choix* n'est optimal, donc il existe des situations où ceux que nous avons proposés dans le TP ne sont pas les meilleurs : il faut savoir s'adapter ! Donnons nos motivations :

- pourquoi de pas avoir utiliser UTF-8 ? L<sup>A</sup>T<sub>E</sub>X n'est pas encore (complètement) prêt ; néanmoins, c'est un encodage universel<sup>45</sup> qui a vocation à s'imposer d'ici quelques

---

43. Coupure de fin de ligne.

44. Ce qui ne change rien pour L<sup>A</sup>T<sub>E</sub>X lors de la compilation, mais facilite la lecture du fichier `.tex`.

45. Cet encodage est commun à tous les systèmes d'exploitation et vise à regrouper l'ensemble des caractères utilisés dans toutes les langues officielles de la Terre.

- années (c'est en général l'encodage par défaut sous Linux) ;
- pourquoi éviter l'encodage Mac OS Roman ? Parce qu'il est vraiment très (trop ?) spécifique à ce système d'exploitation ;
- pourquoi proposer ISO-8859-1 ? Parce que cet encodage (de Linux) est très proche de celui de Windows ;
- pourquoi alors ne pas avoir utilisé ce dernier encodage pendant le TP ? Parce que la version actuelle<sup>46</sup> de `TEXnicCenter` ne permet pas de changer d'encodage ; néanmoins, les différences entre l'ISO-8859-1 et le cp1252 de Windows sont tellement minimales que dans la pratique on ne rencontre pas de problèmes si l'on confond les deux.

Il faut bien comprendre que le choix de l'option du package `inputenc` ne modifie pas l'encodage de l'éditeur ; il permet seulement d'indiquer à  $\LaTeX$  quel encodage est utilisé par l'éditeur (puisque'il ne peut pas le deviner tout seul...). Il appartient donc à l'utilisateur de donner à `inputenc` l'option correspondant aux réglages de l'éditeur.

### B.3.3 Quelques autres packages utiles

*Donnons ici une liste de certains packages (non abordés dans ce TP) qui peuvent se révéler utiles ; cette liste est bien sûr très loin d'être exhaustive ! Pour toute information, se reporter à la documentation du package.*

Il existe un très grand nombre de packages, qui pourraient être classés en trois catégories : certains deviennent plus ou moins incontournables (cf. § B.3.1) d'autres enrichissent des fonctionnalités de  $\LaTeX$  déjà existantes (tels les entêtes et pieds de page) et d'autres encore sont très spécialisés (pour la composition de molécules chimiques par exemple).

Il faut savoir que les packages évoluent au gré de leur(s) concepteur(s) ; ainsi, certains deviennent obsolètes et sont remplacés par des plus performants. Un bon conseil est de vérifier la date de dernière mise à jour du package avant de s'en servir.

**enumitem** pour personnaliser les trois types de listes ; l'option `shortlabel` est souvent celle recherchée par les utilisateurs.

**fancyhdr** pour personnaliser les entêtes et pieds de page.

**floatrow** pour personnaliser les flottants.

**hyperref** pour faciliter la navigation hypertexte dans les versions PDF (*i.e.* transformer les `\ref{}`, `\cite{}`, etc. en liens cliquables) et pour gérer quelques outils offerts par le format PDF (cf. § 4.7.4 de [lshort-fr]).

**listings** pour mettre en forme des lignes de code ; il connaît<sup>47</sup> énormément de langages informatiques. Attention, utiliser des caractères propres à l'UTF-8 à l'intérieur d'un environnement `lstlisting` peut être source d'incompatibilités.

---

46. Il s'agit de la version 1.0, *i.e.* la première version stable ; notons qu'elle est sortie en 2008 alors que cet éditeur était utilisé depuis plusieurs années auparavant : c'était sous des versions dites « bêta ». Notons enfin que la version 2, actuellement disponible en version bêta, permet de gérer d'autres encodages.

47. Cela signifie qu'il est capable de reconnaître les mots-clé, les commentaires, etc. d'un langage et d'adapter la mise en forme en conséquence.

**siunitx** pour gérer d'une manière cohérente l'écriture des unités ; on évitera l'usage de `Slunits...`

**subfig** pour gérer les sous-flottants.

**tocbibind** pour ajouter automatiquement à la table des matières des entrées qui ne s'y mettent pas toutes seules (commandes de sectionnement étoilées, etc.).

**url** pour mieux assurer la césure des adresses web ou email ; attention, la commande `\url{}` est *fragile* (explication de ce mot dans [lshort-fr]).

**varioref** pour améliorer l'utilisation de `\pageref{}` (en écrivant « page suivante » plutôt que le numéro de page si l'on fait référence à ... la page suivante, etc.).

**xspace** pour ne pas avoir à taper des accolades à la fin de certaines commandes si l'on veut une espace derrière (*i.e.* `1\ier exercice` au lieu de `1\ier{} exercice`, etc.).

## B.4 Le préambule et les macros

*Revenons ici sur la deuxième partie d'un préambule classique, introduit à la section 2. Il s'agit d'un outil de personnalisation très puissant de L<sup>A</sup>T<sub>E</sub>X : on peut créer ses propres commandes (voir le paragraphe 6.1.1 de [lshort-fr]).*

Cela se fait avec la commande `\newcommand*{ } [ ] { }` (l'étoile est à retirer si la commande doit s'appliquer à un objet plus grand qu'un paragraphe). Illustrons cette fonctionnalité à travers différents exemples (sauf mention contraire, les extraits de code qui suivent sont à taper *dans le préambule*, après le chargement des différents packages) :

1. Abréger un nom de commande : `\newcommand*\eps{\varepsilon}`. Il suffira alors de taper (dans le corps de document) `\eps` pour afficher  $\varepsilon$ .
2. Distinguer le fond et la forme : `\newcommand*\nom[1]{\textsf{#1}}`. Cet exemple illustre d'ailleurs l'utilisation des commandes à paramètres (un seul ici). C'est une macro utilisée dans cette feuille de TP : dès que l'on a cité un nom de package, d'environnement, de logiciel, etc. on a écrit, par exemple, `\nom{beamer}` au lieu de `\textsf{beamer}`. C'est *a priori* se compliquer la vie, mais il n'en est rien : en distinguant ainsi le fond et la forme dans le fichier source, on le rend plus lisible et plus facile à corriger ; autre avantage : si au dernier moment on doit afficher ces noms en **type machine à écrire** (par exemple pour se coordonner avec un collègue qui aurait rédigé une autre partie du document), il suffit de changer un seul endroit du fichier source et tout le reste en découlera automatiquement. D'autres exemples convaincants sont présents dans le gros exemple (page 19).
3. Créer de nouveaux opérateurs mathématiques (certains opérateurs courants ou francisés ne figurent pas dans L<sup>A</sup>T<sub>E</sub>X) : `\DeclareMathOperator{\divg}{div}` permet de définir l'opérateur divergence ; il permet d'écrire (dans le corps de document) :

Formule de la divergence : $\left[ \oint_{\partial\Omega} \vec{X} \, dS = \int_{\Omega} \operatorname{div} \vec{X} \, dx \right]$	Formule de la divergence : $\oint_{\partial\Omega} \vec{X} \, dS = \int_{\Omega} \operatorname{div} \vec{X} \, dx.$
--	--

Ces deux commandes ne permettent pas de renommer une macro déjà existante (par exemple, `\div` existe sous  $\LaTeX$ ). On remarque aussi que pour utiliser `\eps` définie ci-dessus, on doit être en mode mathématique : en rajoutant la bonne commande dans la définition de cette macro, on peut faire en sorte que  $\LaTeX$  passe automatiquement en mode mathématique quand c'est nécessaire...

Mentionnons pour finir qu'il est possible de personnaliser plein d'autres choses : des environnements, des compteurs, des flottants, ... on renvoie à `[lshort-fr]`.

## B.5 Où trouver de l'aide ?

Si vous avez besoin d'aide, sachez qu'il y a de fortes chances que quelqu'un ait déjà été confronté à votre problème avant vous : une réponse figure donc sans doute dans une quelconque documentation...

- Parmi les polycopiés librement téléchargeables : `[lshort-fr]` distribué en cours (dont la version PDF est régulièrement actualisée sur Internet) ou `[Loz]` ;
- Parmi les livres : le très pédagogique `[B-Ch]` (dont une nouvelle édition est prévue pour juin 2010) ou `[LPI]` ;
- Parmi les FAQ : le PDF en cours de rédaction `[FAQ]`, qui est sans doute la plus complète en français.
- Pour ceux disposant d'un accès permanent à Internet, des FAQ et des forums (pour débutants) en ligne existent : la partie  $\LaTeX$  des sites `mathematex.net` ou `developpez.com` par exemple ;
- Pour les mathématiques, quelques pistes figurent au paragraphe 3.6.3.

Le dernier recours reste d'utiliser son moteur de recherche favori et de s'accrocher pour trouver la réponse la plus pertinente, en prenant garde au fait que la très populaire FAQ de l'équipe Grappa de Lille ne semble plus mise à jour depuis longtemps.

## Références

- [amsl doc] American Mathematical Society. *User's Guide for the `amsmath` Package*, février 2002 (document PDF). <http://www.ctan.org/>.
- [And] Jacques ANDRÉ. *Petites leçons de typographie*, novembre 2008 (document PDF). <http://jacques-andre.fr/>.
- [B-Ch] Denis BITOUZÉ, Jean-Côme CHARPENTIER. *L<sup>A</sup>T<sub>E</sub>X*. Collection Synthex, Pearson Education France, 2006.
- [FAQ] *La FAQ de fr.comp.text.tex*, novembre 2004 (document PDF). <http://faqfctt.fr.eu.org/>.
- [Fli] Daniel FLIPO. *Documentation sur le module frenchb de Babel*, décembre 2008 (document PDF). <http://daniel.flipo.free.fr/frenchb/>.
- [lshort-fr] Tobias OETIKER (pour la version anglaise), traduit en français. *Une courte (?) introduction à L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*, janvier 2010 (document PDF). CTAN://info/lshort/french/.
- [Loz] Vincent LOZANO. *Tout ce que vous avez toujours voulu savoir sur L<sup>A</sup>T<sub>E</sub>X sans jamais oser le demander*, octobre 2008 (document PDF) <http://cours.enise.fr/info/latex/>.
- [LPI] Céline CHEVALIER et co-auteurs. *L<sup>A</sup>T<sub>E</sub>X pour l' impatient*. H&K, 2005.
- [Mathmode] Herbert Voß. *Math mode*, janvier 2009 (document PDF). <http://www.ctan.org/>.
- [McC] McClinews. *Découverte de Beamer*, décembre 2008. <http://mcclinews.free.fr/latex/introbeamer.php>.
- [Obj] Vincent BECK, Jérôme MALICK, Gabriel PEYRÉ. *Objectif Agrégation*, 2<sup>e</sup> édition. H&K, 2005.
- [Ortho] *La nouvelle orthographe, parlons-en !*, septembre 2007 (document PDF). <http://www.orthographe-recommandee.info/>.
- [Pak] Scott PAKIN. *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List*, septembre 2005 (document PDF). <http://www.ctan.org/>.

