

Exposés sur L^AT_EX

Cours 3

Des *packages* incontournables

Thierry Masson

Centre de Physique Théorique
Campus de Luminy, Marseille



Cours 3 – Des *packages* incontournables

- Le *package* **inputenc**
- Le *package* **fontenc**
- Le *package* **textcomp**
- Le *package* **lmodern**
- Le *package* **geometry**
- Le *package* **babel**
- Le *package* **hyperref**
- Les *packages* **amssymb** et **amsmath**
- Le *package* **graphicx**

“Give a man a fish and he will eat for a day;
Teach a man to fish and he will eat for a lifetime.
The moral? Read the manual!”
Sign on a computer system consultant’s desk.

Une modèle de fichier source

Voici à quoi peut ressembler un fichier source \LaTeX de base :

```
\documentclass[12pt]{article}
\usepackage[latin1]{inputenc}
\usepackage[TS1,T1]{fontenc}
\usepackage{textcomp}
\usepackage[a4paper, tmargin=3cm, bmargin=3cm,
             rmargin=2.2cm, lmargin=2.2cm]{geometry}
\usepackage{amssymb, amsmath}
\usepackage{graphicx}
\usepackage{lmodern}
\usepackage[english, french]{babel}
\usepackage{hyperref}

\begin{document}
  Mon texte\dots
\end{document}
```

Ces lignes sont commentées dans ce qui suit, mais pas dans l'ordre...

Ces *packages* peuvent être utilisés avec d'autres classes que la classe **article**.

➔ **report**, **book**, **amsart**, **revtex**, **svjour**...

Le package inputenc

Où l'on prend conscience que notre façon de communiquer avec l'ordinateur est très rudimentaire, mais où l'on se réjouit de découvrir l'existence du Code Universel...

L'encodage du fichier source

En entrée (fichier source), le moteur `tex` accepte seulement un encodage sur 8 bits

→ $2^8 = 256$ caractères possibles seulement.

Deux conséquences importantes :

- 1 Il faut recourir à des macros pour des symboles absents de ces 256 possibilités : `\alpha` → α , `\textcopyright` → ©, et les lettres accentuées peu usuelles... `LATEX` permet actuellement l'accès à plus de 5 000 symboles par des macros diverses (avec des *packages* et polices appropriés).
- 2 Il n'y a pas d'encodage standard sur 8 bits dans le monde informatique.

Avant, il y avait l'`ASCII`[Ⓔ], encodage "standard" sur 7 bits (128 caractères) : les lettres usuelles (minuscules et majuscules), les chiffres, quelques symboles divers (`$`, `%`, `#`, `{`, `}`...). La première version de `tex` fonctionnait sous 7 bits, d'où l'usage de `\'e` pour encoder `é`.

Le choix de compléter ce tableau à 256 caractères dépend des systèmes d'exploitations : Windows (`ansinew`[Ⓔ]), Macintosh (`applemac`[Ⓔ]) et Unix/Linux (`latin1`[Ⓔ]) n'ont pas choisi de placer les nouveaux caractères aux mêmes endroits :

"é" est en position hexadécimale E9, 8E et E9 respectivement ;

"œ" est absent de `latin1`.

Interlude : UNICODE

UNICODE[®] est une norme informatique qui vise à donner à tout caractère de n'importe quel système d'écriture un *nom* et un *identifiant numérique*, et ce de manière unique.

Intérêt : Web, échanges de fichiers, textes contenant différentes langues...

UNICODE est donc un système d'encodage des caractères. Mieux encore, il les définit !

Le standard UNICODE est une réponse à la disparité des encodages :

- chaque caractère est nommé et défini ;
- la place des caractères est standardisée ;
- toutes les écritures (présentes et passées) de la Terre sont incluses (ou le seront) ;
- tous les symboles techniques (mathématique, ingénierie, musique...) ont leur place.

Une police UNICODE est une police qui contient tous les glyphes de tous les caractères définis dans UNICODE, soit environ 70 000 objets !

UNICODE ne cesse ainsi jamais d'évoluer : chacun peut proposer des caractères nouveaux, voire des écritures entières.

Dans une prochaine version, on y trouvera probablement le *Tengwar*[®], l'écriture inventée par J. R. R. Tolkien et qui figure en bonne place dans **Le Seigneur des Anneaux**[®].

UTF-8 est une implémentation concrète et performante de l'encodage UNICODE.

(Voir la page Wikipédia[®] et les **Tableaux de caractères Unicode**[®].)

Interlude : UNICODE (suite)

Extrait de **Introduction à Unicode et à l'ISO 10646** [Ⓜ] par Patrick ANDRIES :

Glyphe représentatif	Numéro	Nom officiel français de l'ISO/CEI 10646
D	U+0044	LETTRE MAJUSCULE LATINE D
v	U+0076	LETTRE MINUSCULE LATINE V
o	U+006F	LETTRE MINUSCULE LATINE O
ř	U+0159	LETTRE MINUSCULE LATINE R CARON
a	U+0061	LETTRE MINUSCULE LATINE A
k	U+006B	LETTRE MINUSCULE LATINE K
	U+0020	ESPACE
天	U+5929	IDÉOGRAMME UNIFIÉ CJC-5929
心	U+2F3C	CLÉ CHINOISE CŒUR
س	U+0633	LETTRE ARABE SÎN
ل	U+0644	LETTRE ARABE LAM
ا	U+0627	LETTRE ARABE ALIF
م	U+0645	LETTRE ARABE MÎM
α	U+03B1	LETTRE MINUSCULE GRECQUE ALPHA
ϴ	U+10338	LETTRE GOTIQUE TH
♪	U+1D110	SYMBOLE MUSICAL POINT D'ORGUE
	U+2270	NI PLUS PETIT NI ÉGAL À
-ལ་མ་	U+0F06	SIGNE TIBÉTAIN D'INSERTION YIG MGO PHUR SHAD MA ¹⁵

`inputenc` et l'encodage du fichier source

Le package `inputenc` permet de désigner l'encodage sélectionné pour le code source, c'est à dire celui que l'éditeur de texte utilise pour la sauvegarde :

`\usepackage[latin1]{inputenc}` sélectionne l'encodage `latin1`.

Aujourd'hui, les éditeurs de texte savent interpréter différents encodages.

S'il est accessible, il est souhaitable d'utiliser `UNICODE` qu'on peut sélectionner avec :

`\usepackage[utf8]{inputenc}`

Remarque : l'encodage d'entrée n'influence pas sur le traitement du texte par `LATEX`.

`\'e` donne le même résultat que `é`.

Mais des correcteurs orthographiques auront du mal à corriger `h\'eb\'et\'e` !

Remarque : il arrive souvent qu'on reçoive un fichier source dans un encodage qui n'est pas celui de notre éditeur de texte.

À l'édition, le fichier semble rempli de hiéroglyphes bizarres !

È 1'Ždition, le fichier semble rempli de hižroglyphes bizarres !

(encodage `applemac` ouvert en `ansinew`)

Si `inputenc` est appelé avec la bonne option, `LATEX` compilera ce fichier sans soucis.

Remarque : il est possible d'installer plusieurs encodages pour un document et de mélanger des textes en entrée (avec des `\input` ou `\include{-}`) de différents encodages : placer `\inputencoding{encodage}` à chaque changement d'encodage.

Le *package* fontenc

Où l'on prend la mesure des limitations de nos ordinateurs programmés en 8 bits, et où l'on s'applique à dépasser avec élégance et efficacité ces contingences...

L'encodage des fontes

Les fichiers de fontes `.pfb` utilisés par `LATEX` ne contiennent que 256 glyphes (8 bits...). Pour composer un texte un peu riche, il faut donc assembler de nombreux fichiers `.pfb`.

Les polices d'écriture par défaut de `TEX`, `COMPUTER MODERN`, contiennent 90 fichiers `.pfb` pour autant de fontes de caractères.

Deux problèmes se présentent à `LATEX` :

- ① Il doit trouver les informations sur un glyphe particulier dans cette multitude de fichiers, et son emplacement dans les fichiers (qui sont en fait des gros tableaux).
- ② Il doit construire un fichier de sortie (`.dvi`, `.pdf`) dans lequel l'information est du type : à tel endroit de la page placer le glyphe numéro *tant* pris dans le fichier *untel*.

Pour trouver un glyphe particulier, `LATEX` doit d'abord connaître le fichier dans lequel il se trouve (c'est le rôle des fichiers `.fd`, `.vf`...) et ensuite trouver l'emplacement du glyphe dans le tableau des 256 caractères.

Pour placer la bonne information dans le fichier de sortie, il doit connaître l'organisation du tableau des 256 caractères.

“L'encodage des fontes” standardise la façon dont on morcelle une police d'écriture en blocs “réduits” de 256 glyphes.

Le package `fontenc` informe `LATEX` sur l'encodage utilisé par la police utilisée.

Une même police peut être morcellée dans différents encodages ➡ choix de l'utilisateur.

Les encodages de fontes usuels

Les encodages utiles pour le commun des mortels de langue latine :

OT1 C'est l'encodage d'origine, "O" pour "Old"...

Cet encodage ne reconnaît pas les glyphes des lettres accentuées.

À éviter car il ne gère pas les césures sur des mots contenant des caractères accentués.

T1 Il contient l'essentiel des caractères des langues de l'Europe de l'ouest.

Il permet la gestion des césures de la majorité de ces langues.

C'est l'encodage à utiliser de préférence.

TS1 Encodage "Text Symbol". C'est un encodage compagnon de **T1** qui permet l'accès à des glyphes de type "symboles" utilisés dans des textes, comme ©, ‰ ou †.

OML, OMS, OMX Encodages utilisés pour les polices mathématiques. Il est inutile de les spécifier dans **fontenc** car ils sont automatiquement appelés.

Il existe des encodages spécifiques pour le cyrillique, le grec, le chinois/japonais/koréen (CJK) et même le phonétique !

Pour utiliser les encodages **T1** et **TS1**, la ligne à placer dans le préambule est du type :

```
\usepackage[TS1,T1]{fontenc}
```

L'importance du choix de l'encodage des fontes

Dans l'encodage **OT1**, les glyphes des lettres accentuées n'existent pas.

Dans ce cas, **tex** superpose l'accent (dont le glyphe existe) sur la lettre correspondante.

Le résultat est double :

- 1 Le glyphe obtenu n'est pas optimum d'un point de vue typographique.
- 2 La métrique du glyphe ainsi composé n'existe pas, ce qui rend inopérant l'algorithme de césure utilisé par **tex**.

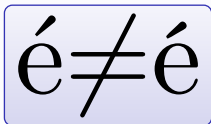
Pour résoudre ces problèmes, il suffit de passer en encodage **T1** (pour les langues latines).

Les polices d'écriture par défaut de **T_EX** sont les déclinaisons de **COMPUTER MODERN (CM)**. Elles ont été créées par D. Knuth en même temps que **T_EX**. Ces polices ne peuvent pas être utilisées avec l'encodage **T1** car elles ne contiennent pas suffisamment de glyphes.

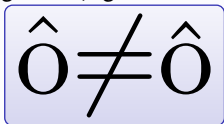
Pour pouvoir utiliser l'encodage **T1**, de nouvelles polices ont été créées, **CM-SUPER**, qui contiennent les glyphes des lettres accentuées communes.

Ces polices sont activées par défaut lorsqu'on passe en **T1**.

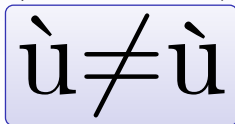
Les glyphes obtenus étant différents entre ces deux ensembles de polices, le résultat visuel peut différer si on utilise l'encodage **OT1** (à gauche, **CM**) ou **T1** (à droite, **CM-SUPER**) :



é ≠ é



ô ≠ ô



ù ≠ ù

Des packages incontournables

L'importance du choix de l'encodage des fontes (suite)

⚠ L'algorithme de césure utilisé par `tex` dépend de l'encodage des polices.
 ➔ très important pour l'utilisateur.

Avec l'encodage T1

Il ôta son chapeau
 devant le médecin hébété.
 Il ôta son chapeau devant le mé-
 decin hébété.

Avec l'encodage OT1

Il ôta son chapeau
 devant le médecin hébété.
 Il ôta son chapeau devant le
 médecin hébété.

Avec l'encodage T1

Il \^ota son chapeau
 devant le m\'edecin
 h\'eb\'et\'e.
 Il ôta son chapeau devant le mé-
 decin hébété.

Avec l'encodage OT1

Il \^ota son chapeau
 devant le m\'edecin
 h\'eb\'et\'e.
 Il ôta son chapeau devant le
 médecin hébété.

Noter les césures différentes entre la colonne de gauche (T1) et la colonne de droite (OT1), indépendamment de la façon dont on compose le texte.

⚠ Autre problème de l'encodage OT1 : impossible de rechercher des mots avec des lettres accentuées dans le PDF final.

Les packages `inputenc` et `fontenc` : la synthèse

Entre l'entrée du fichier source dont l'encodage est repéré par `inputenc`, et la sortie "visuelle" dont l'encodage des polices est imposée par `fontenc`, le moteur `tex` utilise son propre encodage !

C'est pourquoi il est important qu'il sache traduire le texte en entrée dans son propre encodage et la sortie dans l'encodage des polices utilisées.

Après réglage correct de l'éditeur de texte utilisé pour composer le fichier source (préférence d'encodage de sauvegarde), systématiquement prendre l'une des options suivantes :

```
\usepackage[latin1]{inputenc}
\usepackage[TS1,T1]{fontenc}
```

ou

```
\usepackage[utf8]{inputenc}
\usepackage[TS1,T1]{fontenc}
```

 Il y a une subtilité sur le choix des polices d'écriture, voir le package `lmodern`.

Le *package* textcomp

Où l'on découvre que l'alphabet n'est pas tout et qu'il est aussi possible de communiquer avec des signes bizarres...

textcomp : text companion

Pour faire de la place aux glyphes des alphabets latins dans l'encodage T1, il a fallu placer les symboles divers (autres que mathématiques) dans d'autres fontes, avec l'encodage TS1 :

£ ¤ § ¶ † © ® % ¯ ‡ ▪ * ™ 

Certains de ces symboles sont déjà accessibles sans **textcomp**.

`\usepackage{textcomp}` permet à L^AT_EX d'accéder à ces symboles divers.

`\texteuro`, `\texttrademark` et `\textsection`
 €, ™ et §



mathcomp est la version "math" de **textcomp** : certains symboles de **textcomp** y sont redéfinis en mode mathématique, avec le préfixe `\tc-` plutôt que `\text-`.

`$45\tcdegree$`, `$100\,\tcohm$` et `$12\tccelsius$`
 45°, 100 Ω et 12°C

⚠ **mathcomp** n'utilise pas nécessairement la police du texte par défaut, il faut lui désigner cette police comme option.

textcomp : ¥€\$ OR №?

Exemple de symboles accessibles avec **textcomp** (126 symboles en tout) :

<code>\textasteriskcentered</code>	*	<code>\textdollar</code>	\$
<code>\texteuro</code>	€	<code>\textyen</code>	¥
<code>\textasciicircum</code>	^	<code>\textasciitilde</code>	~
<code>\textbar</code>		<code>\textbackslash</code>	\
<code>\textbraceleft</code>	{	<code>\textbraceright</code>	}
<code>\textlbrackdbl</code>	[[<code>\textrbrackdbl</code>]]
<code>\textrightarrow</code>	→	<code>\textleftarrow</code>	←
<code>\textdownarrow</code>	↓	<code>\textuparrow</code>	↑
<code>\textlangle</code>	<	<code>\textrangle</code>	>
<code>\textcopyright</code>	©	<code>\textregistered</code>	®
<code>\texttrademark</code>	™	<code>\textservicemark</code>	SM
<code>\textdagger</code>	†	<code>\textdaggerdbl</code>	‡
<code>\textperthousand</code>	‰	<code>\textpertenthousand</code>	‱
<code>\textdegree</code>	°	<code>\textcelsius</code>	°C
<code>\textborn</code>	*	<code>\textmu</code>	μ
<code>\textohm</code>	Ω	<code>\textmho</code>	℧
<code>\textleaf</code>		<code>\textmusicalnote</code>	
<code>\textmarried</code>	∞	<code>\textdivorced</code>	∅

Le *package* Tmodern

Où l'on s'étonne que l'Amérique ait si longtemps ignoré la diversité et la richesse typographique des langues latines, mais où l'on apprend que l'Europe Unie s'est mobilisée pour réparer cette maladresse...

Les polices LATIN MODERN

Les polices par défaut de **T_EX**, **COMPUTER MODERN**, sont insuffisantes d'un point de vue typographique, par manque de nombreux glyphes (lettres accentués pour l'essentiel). Les polices **CM-SUPER** ont complété cette lacune provisoirement.

Depuis 2002, un projet financé principalement par des groupes d'utilisateurs **T_EX** européens, a vu le jour pour étendre **COMPUTER MODERN** à toutes les écritures basées sur le latin, en particulier les langues d'Europe de l'Est.

De nombreux glyphes ont été ajoutés, la qualité typographique a été renforcée.

Le résultat est l'ensemble de polices d'écriture **LATIN MODERN**, composé de 72 fontes de caractères, auxquelles on ajoute 20 fontes pour les mathématiques.

En tout, il y a 58 604 caractères pour un total d'environ 69 000 glyphes...

`\usepackage{lmodern}` permet d'utiliser ces polices.

Tous les fichiers nécessaires sont installés par défaut sur les dernières versions de **T_EX**. Il est hautement recommandé de l'utiliser.

Exemples de glyphes contenus dans les polices **LATIN MODERN** :

Ÿ ð Ê Þ Ę Ń á À Ą Ę

(Extrait de **An exploration of the Latin Modern fonts** TM par Will ROBERTSON)

Les avantages des polices LATIN MODERN

⚠ Il y a des différences nettes lorsqu'on agrandit les polices `LATIN MODERN`, `CM-SUPER` et `COMPUTER MODERN` avec des commandes usuelles :

```
\bfseries\normalsize
```

Latin Modern Sans Bold
CM-super Sans Bold
Computer Modern Sans Bold

```
\bfseries\large
```

Latin Modern Sans Bold
CM-super Sans Bold
Computer Modern Sans Bold

Les polices `CM-SUPER` ne changent pas d'échelle correctement et produisent du "Sans Bold" qui est trop maigre.

On a vu qu'il était préférable d'utiliser l'encodage `T1` avec `fontenc` pour avoir de bonnes césures ➔ ceci oblige `LATEX` à utiliser les polices `CM-SUPER` qui n'ont pas de grandes qualités typographiques.

Utiliser systématiquement `lmodern`, ces polices ont tous les avantages.

Enfin, les polices `LATIN MODERN` sont disponibles en `OPENTYPE`, ce qui les rend accessibles à n'importe quel logiciel "ordinaire" sur l'ordinateur.

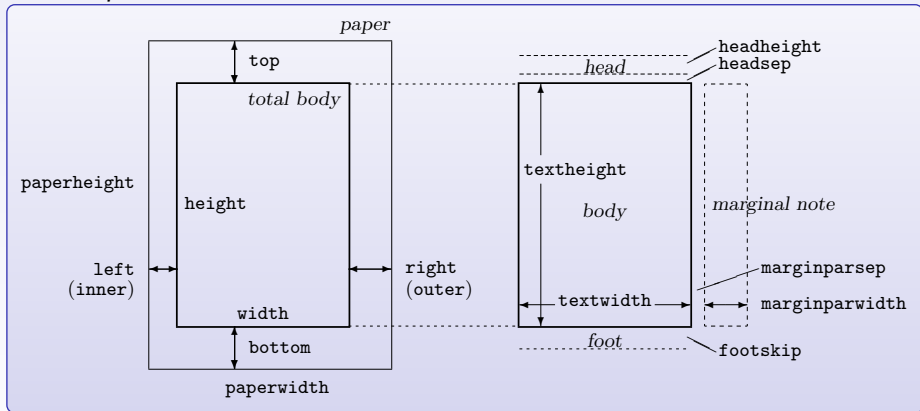
➔ Pratique pour externaliser la production d'un morceau de document tout en conservant la typographie de `TEX`.

Le package geometry

Où l'on découvre comment maîtriser l'espace blanc de la page et où l'on apprend à coucher notre prose à l'endroit de notre choix...

La géométrie d'une page L^AT_EX

Une page L^AT_EX est divisée en zones précises, dont la géométrie est gouvernée par différents paramètres :



Il est possible de fixer soi-même ses paramètres, mais il faut alors respecter certaines contraintes comme par exemple :

$$\text{paperwidth} = \text{left} + \text{width} + \text{right}$$

$$\text{paperheight} = \text{top} + \text{height} + \text{bottom}$$

Gestion de la géométrie avec le *package geometry*

Le *package geometry* permet de gérer cette situation complexe.

Il donne à l'utilisateur la possibilité de fixer les valeurs de certains paramètres, et il se charge de calculer ceux qui manquent.

- En cas de sous-détermination, il utilise des valeurs par défaut.
- En cas de sur-détermination, il y a des règles de priorité.
- Il est possible de sauvegarder une géométrie, de basculer sur une nouvelle géométrie (qui démarre sur une nouvelle page) et d'appeler une géométrie sauvée.
- Il est possible de choisir une zone de travail (`\layout`) plus petite que la taille du papier.

La géométrie de la page peut être fixée par :

- Les options dans `\usepackage[-]{geometry}`
- La commande `\geometry{-}` placée dans le préambule.
Il faut bien sûr appeler le *package* avant d'utiliser cette commande.
Cette commande est prioritaire sur les options du *package*.

Les options spécifiées avec ce *package* sont prioritaires sur celles de même nature fournies avec la classe du document (taille du papier).

Exemple de géométrie

```
\usepackage[a4paper, tmargin=3cm, bmargin=3cm,
             rmargin=2.2cm, lmargin=2.2cm]{geometry}
```

Spécifie : la taille du papier (`a4paper`), les marges “top”, “bottom”, “right” et “left” (`tmargin`, `bmargin`, `rmargin` et `lmargin`).

C'est équivalent à :

```
\usepackage[a4paper, vmargin=3cm, hmargin=2.2cm]{geometry}
```

où `vmargin` et `hmargin` sont les marges “verticales” et “horizontales”.

Quelques options utiles :

Options générales `verbose`, `twoside`, `showframe`, `landscape`, `portrait`...

Taille du papier `a0paper`, ... , `a6paper`, `screen`, `paperwidth`= $\langle dim \rangle$,
`paperheight`= $\langle dim \rangle$...

Taille du corps `textwidth`= $\langle dim \rangle$, `textheight`= $\langle dim \rangle$, `text`={ $\langle dim \rangle$, $\langle dim \rangle$ }...

Taille des marges `tmargin`= $\langle dim \rangle$, ... , `hmargin`= $\langle dim \rangle$, `vmargin`= $\langle dim \rangle$,
`inner`= $\langle dim \rangle$, `outer`= $\langle dim \rangle$...

➔ Cette panoplie d'options permet à l'utilisateur de définir des marges ou au contraire une taille du corps du texte.

➔ Gestion des marges intérieures et extérieures pour les impressions en recto-verso.

Le package babel

Où l'on se rassure d'apprendre que l'anglais n'est pas la langue unique par laquelle nous aurons à communiquer et où l'on découvre qu'au contraire nos incantations peuvent convoquer les règles typographiques de nombreuses langues...

Au tour de babel

LA_TE_X utilise par défaut la typographie de la langue anglaise (américaine).

Les mots clés structurants (“Chapter”, “Contents”, “Table”...) sont aussi en anglais.

Le package **babel** permet de gérer les spécificités de plus de 40 langues :

- traduction des mots usuels produits par **L**A_TE_X ainsi que de certaines macros ;
- utilisation des motifs de césure et des règles typographiques de la langue spécifiée ;
- lot de macros spécifiques à la langue.

```
\usepackage[english,french]{babel}
```

installe les langues “english” et “french” pour le document et sélectionne “french” comme langue par défaut (le dernier de la liste).

Il faut placer dans cette liste toutes les langues utilisées par la suite dans le document.

`\selectlanguage{english}` active les attributs de la langue anglaise.

`\begin{otherlanguage}{german} . . . \end{otherlanguage}`

permet d’englober un morceau de texte en allemand.

babel définit de nombreuses commandes pour gérer plusieurs langues dans les macros personnelles ➔ lire sa documentation pour des détails.



Il faut placer l’appel à **babel** après les *packages* susceptibles de définir des mots clés.

babel et les spécificités de la langue française

Traduction des mots usuels : “Chapter” → “Chapitre”, “Contents” → “Table des matières”, “Table” → “Tableau”...

Date : `\today` donne la date en français :

anglais → February 25, 2012

français → 25 février 2012

Typographie française : Indentation pour le premier paragraphe,

guillemets français : `\og mot \fg` → « mot »,

espace avant les doubles ponctuations : ; ! ?

Macros diverses : `\primo` → 1^o, `\secundo` → 2^o, `M\up{me}` → M^{me}, `1\ier` → 1^{er}, `3\ieme` → 3^e, `\Nos` → Nos ...

Listes : Les items des listes sont “-” au lieu de “.” et les espaces sont ajustés à la typographie française.

Césures : Les césures françaises sont différentes des césures anglaises :

En anglais, `\showhyphens{signal container}` produit

[] \T1/lmr/m/n/12 sig-nal con-tainer

En français, `\showhyphens{signal container}` produit

[] \T1/lmr/m/n/12 si-gnal contai-ner

→ `tex` propose de couper ces mots à des endroits différents.

Le *package* hyperref

Où l'on découvre qu'un document écrit peut se révéler dynamique sous les cliquetis d'une simple souris...

Hyperliens, navigation et métadonnées avec PDF

Le format **PDF** a des fonctionnalités avancées :

hyperliens internes qui permettent de passer d'une page à une autre dans le document ;

hyperliens externes qui acceptent les **URL** usuelles, par exemple vers des pages web (`http:// . . .`) ou des adresses emails (`mailto: . . .`) ;

table des matières qui permet de naviguer dans le document (*bookmarks*) ;

métadonnées du document qui renseignent sur l'auteur, le titre, les mots clés, le logiciel de création...

Le package **hyperref** permet de profiter quasi-automatiquement de ces fonctionnalités :

- hyperliens internes vers les `\label{-}` à partir de `\ref{-}`, `\pageref{-}` et `\eqref{-}` ;
- hyperliens internes à partir de la table des matières **L^AT_EX** (`\tableofcontents`) vers les pages concernées, de même pour les listes de figures et de tableaux ;
- hyperliens internes vers les entrées bibliographiques à partir de `\cite{-}` ;
- table des matières du document **PDF** générée sur les informations `\chapter{-}` ... `\subsection{-}` du document.

Le package **hyperref** est compatible avec `pdftex`, `tex/dvips/ps2pdf` et `tex/dvipdfm`.

Configuration de hyperref, métadonnées du PDF

La commande `\hypersetup{-}` permet de personnaliser le comportement de **hyperref** et du document **PDF** produit.

```
\hypersetup{
  plainpages=false,
  colorlinks=true, linkcolor=black, anchorcolor=black,
  citecolor=black, menucolor=black, urlcolor=black,
  bookmarks=true, bookmarksopen=true, bookmarksnumbered=true,
  pdftitle={Exposé sur LaTeX},
  pdfauthor={Thierry Masson},
  pdfsubject={cours sur LaTeX},
  pdfcreator={TeX}, pdfproducer={pdfTeX},
  pdfkeywords={LaTeX, cours}
}
```


➔ possibilité d'ouvrir automatiquement le document en pleines pages, couleurs de différents types de liens, comportement des *bookmarks*, métadonnées sur le document...
De nombreuses autres options disponibles...

Quelques commandes utiles

hyperref fournit les commandes essentielles suivantes :

- `\href{http://www.lesite.fr}{lien}` crée un hyperlien externe à partir du texte “lien” vers la page `http://www.lesite.fr`.
- `\hypertarget{label}{lieu}` crée un but de nom “label” à l’emplacement du texte “lieu”.
`\hyperlink{label}{lien}` crée un lien interne vers le but “label” à partir du texte “lien”.
- `\hyperref[label]{lien}` crée un lien interne vers le but créé par `\label{label}`.
- `\texorpdfstring{nom TeX}{nom PDF}`
Les *bookmarks* de PDF ne peuvent pas contenir des macros \TeX .
Cette commande donne une alternative “nom PDF” à “nom TeX”.
Elle doit être utilisée dans les commandes qui produisent les *bookmarks*, comme par exemple `\section{-}`.

L’apparence d’un lien peut être modifiée par des commandes $\text{L}\text{A}\text{T}\text{E}\text{X}$, et par les options de `\hypersetup{-}` (pour la couleur, la présence d’un cadre...).

 Il est préférable d’appeler ce *package* en dernier car il modifie de nombreuses macros définies par d’autres *packages*.

Les *packages* amssymb et amsmath

Où l'on se donne un avant goût de l'esthétisme de la typographie mathématique, mais où l'on reporte à plus tard le plaisir de s'initier plus complètement à cette science...

Mathématiques supérieures avec amssymb et amsmath

Par défaut, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ s'occupe très bien des mathématiques.

De nombreuses fonctionnalités sont ajoutées par ces deux *packages*.

Le *package* **amssymb** définit de nouveaux symboles :

$\backslash\text{varnothing}$ $\rightarrow \emptyset$, $\backslash\text{bigstar}$ $\rightarrow \star$, $\backslash\text{boxplus}$ $\rightarrow \boxplus$, $\backslash\text{ltimes}$ $\rightarrow \ltimes$.

Remarque : d'autres *packages* ajoutent aussi des symboles mathématiques.

amssymb appelle aussi le *package* **amsfonts** qui définit les commandes :

$\backslash\text{mathfrak}\{-}$	o12...789 ABCD...UVWXZ} abc d...u v w x y z
$\backslash\text{mathbb}\{-}$	ABCD...UVWXYZ (⚠ pas de minuscules ni de chiffres)

Le *package* **amsmath** ajoute de nombreuses fonctionnalités :

- des environnements d'équations hors paragraphes : **equation**, **align**, **gather**, **multline**, **split**, **flalign** et des versions étoilées (sans numérotation) ;
- des commandes pour gérer les numéros d'équations, interrompre des équations, placer du texte dans le mode mathématique, des points (à différentes hauteurs), des flèches avec labels, des flèches extensibles, des fractions, des intégrales multiples...
- divers environnements pour gérer des sous équations et des matrices...
- des commandes pour définir des opérateurs (du type $\backslash\text{lim}$).

⚠ Ces deux *packages* ne sont pas réservés aux mathématiciens : il peuvent être très utiles dans un article de physique, et ils sont acceptés par les éditeurs de physique.

Le `package graphicx`

Où l'on apprend qu'on peut faire usage du vieil adage qui prétend qu'un bon dessin vaut mieux qu'un long discours...

L'inclusion de graphiques

`LATEX` ne gère pas directement les fichiers graphiques externes.

C'est le *driver* qui convertit en `PS` ou `PDF` qui s'en occupe.

➔ Selon le *driver* utilisé, les fichiers qu'il est possible d'inclure sont différents :

`dvips` accepte les `.eps`,

`dvipdfm` accepte les `.eps`, `.jpg` et `.png`,

`pdftex` accepte les `.pdf`, `.jpg` et `.png`.

La délégation du travail vers les *drivers* est réalisée grâce au package `graphicx`.

La commande principale définie par ce *package* est `\includegraphics[-]{-}`.

L'argument obligatoire est le nom du fichier image, l'argument optionnel permet des transformations élémentaires : taille finale, rotation, extrait, mise à l'échelle...

```
\includegraphics[width=1.5cm,angle=45]{CPT.pdf}
```



D'autres commandes sont mises à disposition :

- agrandissement d'une boîte `LATEX` quelconque (du texte, une image, un tableau...):
`\scalebox{-}{-}`, `\resizebox{-}{-}{-}`
- rotation d'une boîte `LATEX` : `\rotatebox{-}{-}`

La gestion et la création des graphiques dans `LATEX` fera l'objet d'un autre cours.