
Exposés sur L^AT_EX

2010 – 2011

Thierry MASSON

✉ CPT (UMR 6207)
Case 907 - Campus de Luminy
F-13288 Marseille Cedex 9
✉ `thierry.masson@cpt.univ-mrs.fr`
🌐 `science.thilucmic.fr`

Version du 25 février 2012

Table des matières

1	Les fondamentaux : l'univers T _E X	3
	Préambule en guise d'introduction – La typographie, l'informatique et la place de T _E X – L'installation de T _E X et son écosystème	
2	Les fondamentaux : le document source	23
	Le document source : les bases – Structure du document source – Repères sur les concepts de programmation – Tour d'horizon de commandes de base	
3	Des <i>packages</i> incontournables	45
	Le <i>package</i> inputenc – Le <i>package</i> fontenc – Le <i>package</i> textcomp – Le <i>package</i> lmodern – Le <i>package</i> geometry – Le <i>package</i> babel – Le <i>package</i> hyperref – Les <i>packages</i> amssymb et amsmath – Le <i>package</i> graphicx	
4	La police a bon caractère	61
	Les commandes de base – Commandes plus avancées – Autres polices d'écriture et symboles divers – Les commandes de police en mathématique	
5	Les mathématiques	79
	Généralités – Les <i>packages</i> amsmath et mathtools – Les <i>packages</i> d'environnements de type théorèmes – Quelques <i>packages</i> en vrac	
6	Des <i>packages</i> pour sauver les apparences	103
	Des outils techniques utiles – La gestion de la couleur – La gestion des tableaux – La gestion des listes – La mise en page – Autres éléments du document	
7	L ^A T _E X et les graphiques	121
	Généralités sur les graphiques – Faire des dessins à la souris – Illustrer les mathématiques – Programmer des graphiques – Insertion de graphiques externes – Insertion d'une figure avec légende	
8	Gérer la bibliographie	155
	La gestion manuelle de la bibliographie – Une base de données bibliographiques – Personnaliser les citations – Gérer sa base bibliographique – L'avenir avec bibtex	
9	La classe beamer	179
	Un tour d'horizon des possibles – Les commandes de base – Modifier l'aspect de sa présentation – Au delà de la présentation	
10	Synthèse	203
	Récapitulatif – L'avenir	
	URLs des liens cités dans le texte	211

Cours 1 – Les fondamentaux : l’univers T_EX

« Les majuscules sont des coups de chapeau calligraphiques. »
Louis Jouvét

1.1	Préambule en guise d’introduction	4
	Où l’on découvre ce que D. Knuth nous a apporté, aussi bien dans l’art séculaire de la typographie scientifique qu’en matière de basket-ball . . .	
1.2	La typographie, l’informatique et la place de T _E X	7
	Où l’on apprend que la typographie est un métier de l’art, et où l’on se surprend à croire que l’informatique pourrait remplacer l’artiste . . .	
1.3	L’installation de T _E X et son écosystème	13
	Où l’on apprend comment installer sur son ordinateur le nécessaire et le suffisant, et où l’on essaie d’appréhender la structure nébuleuse des dossiers ainsi confiés à notre bienveillance . . .	

1.1 Préambule en guise d'introduction

Partir du bon pied

Voici une liste très réduite de documents de base sur L^AT_EX^{⌚1}.

Livres : quelques livres très bien faits.

L^AT_EX Companion, F. Mittelbach et M. Goossens, Addison-Wesley 2004.

L^AT_EX, Apprentissage, guide et référence, B. Desgraupes, Vuibert 2003.

Guide to L^AT_EX, H. Kopka et P. Daly, Addison-Wesley 2003.

L^AT_EX pour l'impatient, W. Appel, C. Chavalier, E. Cornet, S. Desreux, H&K, 2009.

Documents électroniques : quelques PDF à lire.

The Not So Short Introduction to L^AT_EX^{⌚2}, un fascicule à conserver sous la main.

Comprehensive L^AT_EX symbol list^{⌚3}, la référence pour trouver un symbole.

Sites internet : quelques liens pour rester à jour.

L^AT_EX Reference Manual^{⌚4}, la référence de toutes les commandes...

Comprehensive T_EX Archive Network^{⌚5}, des *packages* et encore des *packages*...

L^AT_EX Wikibook^{⌚6}, documentation collective sur L^AT_EX, disponible aussi en PDF.

The PracT_EX Journal^{⌚7}, un journal réalisé par des utilisateurs de L^AT_EX.

L^AT_EX Community^{⌚8} et T_EX blog^{⌚9}, pour se tenir au courant de l'actualité.

Documentation en ligne sur L^AT_EX^{⌚10}, ma propre page web de liens.

D'autres références plus spécialisées sont données dans l'exposé.

S'installer confortablement

Voici une liste de logiciels pour installer L^AT_EX.

🐧 Installation fondamentale : **TeXLive^{⌚11}**.

Logiciels d'interfaces : **TeXworks^{⌚12}**, **TeXMaker^{⌚13}**, **emacs^{⌚14}** (+ **AucTeX^{⌚15}**).

Utilitaires divers : **JabRef^{⌚16}** (gestion bibtex), **Aspell^{⌚17}** (correcteur orthographique).

🍏 Installation fondamentale : **MacTeX^{⌚18}**.

Logiciels d'interfaces : **TeXShop^{⌚19}**, **TeXworks^{⌚12}**, **TeXMaker^{⌚13}**, **Aquamacs^{⌚20}** (+ **AucTeX^{⌚15}**).

Utilitaires divers : **BibDesk^{⌚21}** (gestion bibtex), **TeX Live Utility^{⌚22}** (gestion TeXLive), **CocoAspell^{⌚23}** (correcteur orthographique).

🐧 Installation fondamentale : **proTeXt^{⌚24}** (\simeq **MikTeX^{⌚25}**).

Logiciels d'interfaces : **TeXworks^{⌚12}**, **TeXMaker^{⌚13}**, **TeXnicCenter^{⌚26}**, **WinEdit^{⌚27}**, **emacs^{⌚14}** (+ **AucTeX^{⌚15}**).

Utilitaires divers : **JabRef^{⌚16}** (gestion bibtex).

Pendant que ces logiciels s'installent, une récréation intitulée :

“D. Knuth^{⌚28} (créateur de T_EX) : du basket-ball aux panneaux de signalisation”



The Electronic Coach^{⌚29},
An Earthshaking Announcement^{⌚30},
Diamond Signs^{⌚31}.

La typographie scientifique en images

Princeton University Press, 1951

$= \phi(x, g g^{-1} p(b)) = b.$
Define $r_j: P^{-1}p^{-1}(V_j) \rightarrow G$ by $r_j = \beta_j q$. Then
 $r_j \psi_j(b, g) = \beta_j q(\cdot \cdot \cdot, \cdot \cdot \cdot) = \beta_j \tilde{\phi}_j(x, g) = g.$
This proves that ψ_j is a product representation. A similar calculation shows that
 $r_j \psi_j(b, g) = g_j(x) g,$
and therefore the coordinate transformations are $g_\mu(x) \tilde{g}_\mu(p(b))$. The same calculation yields
 $\beta_j q \psi_{j, \mu}(g) = g_\mu(x) g,$
and therefore q is a bundle mapping.
8.13. Associated maps. Let $\mathcal{G}, \mathcal{G}'$ be bundles having the same fibre and group and let h be a map $\mathcal{G} \rightarrow \mathcal{G}'$. The mapping transformations $\{g_\mu\}$ of h are as defined in §2.5. Let $\mathcal{G}, \mathcal{G}'$ be the associated principal bundles. According to §2.6, there is a unique map
 $\tilde{h}: \mathcal{G} \rightarrow \mathcal{G}'$
having the mapping transformations $\{g_\mu\}$. We call \tilde{h} the associated map of the principal bundles.
THEOREM. If h is a map $\mathcal{G} \rightarrow \mathcal{G}'$ and \tilde{h} is the associated map $\mathcal{G} \rightarrow \mathcal{G}'$, then
 $P'(\tilde{h}(\tilde{b}), y) = hP(\tilde{b}, y), \quad \tilde{b} \in \tilde{B}, y \in Y,$
where P, P' are the related principal maps.
If $x = \tilde{p}(\tilde{b}) \in V_\mu$ and $x' = \tilde{h}(x) \in V'_\mu$, then, by (20) of §2.6, we have
 $\tilde{p}'_i \tilde{h}(\tilde{b}) = \tilde{p}'_i \tilde{\phi}'_i(x', \tilde{g}_\mu(x) \tilde{p}_i(\tilde{b})) = \tilde{g}_\mu(x) \tilde{p}_i(\tilde{b}). \quad \dagger$
Applying the definitions of P, P' we obtain the result:
 $P'(\tilde{h}(\tilde{b}), y) = \phi'_i(x', [\tilde{p}'_i \tilde{h}(\tilde{b})] y) = \phi'_i(x', \tilde{g}_\mu(x) \tilde{p}_i(\tilde{b}) y) = h \phi_i(x, \tilde{p}_i(\tilde{b}) y) = hP(\tilde{b}, y).$
The intuitive content of the theorem is based on the interpretation of \tilde{b} as an admissible map $Y \rightarrow Y_\mu$. Then \tilde{b} followed by $h_\mu: Y_\mu \rightarrow Y'_\mu$ is an admissible map $Y \rightarrow Y'_\mu$ and is therefore an element $\tilde{h}(\tilde{b})$ in $G_{x'}$.

Dunod, 1962

du groupe \mathfrak{H} .
b) Le sous-groupe Z . — Désignons par Z l'ensemble des matrices de la forme (2)
$$z = \begin{pmatrix} 1 & 0 \\ z & 1 \end{pmatrix} \quad (3)$$

De la relation
$$z z' = \begin{pmatrix} 1 & 0 \\ z & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ z' & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ z + z' & 1 \end{pmatrix} \quad (4)$$

on conclut que $z z'$ est une matrice de même forme et Z est un sous-groupe du groupe \mathfrak{H} .
2. La décomposition canonique des éléments du groupe \mathfrak{H}
Lemme. — Tout élément $a \in \mathfrak{H}$ satisfaisant à la condition $a_{22} \neq 0$ peut se représenter, et d'une seule manière, sous la forme
 $a = k z, \quad k \in K, z \in Z. \quad (1)$
Démonstration. — L'égalité (1) signifie que
 $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \lambda^{-1} & \mu \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} 1 & 0 \\ z & 1 \end{pmatrix} = \begin{pmatrix} \lambda^{-1} + \mu z & \mu \\ \lambda z & \lambda \end{pmatrix} \quad (2)$
c'est-à-dire que
 $\lambda = a_{22}, \quad \lambda z = a_{21}, \quad \mu = a_{12}, \quad \lambda^{-1} + \mu z = a_{11}. \quad (3)$
Les trois premières égalités donnent
 $\lambda = a_{22}, \quad z = a_{21}/a_{22}, \quad \mu = a_{12}. \quad (4)$
de sorte que k et z sont univoquement définies par l'égalité (1). Réciproquement, si on choisit k et z conformément à la formule (4), l'égalité $\lambda^{-1} + \mu z = a_{11}$ dans (3) est automatiquement satisfaite puisque $a_{11} a_{22} - a_{12} a_{21} = 1$ et donc
 $\lambda^{-1} + \mu z = \frac{1}{a_{22}} + a_{12} \frac{a_{21}}{a_{22}} = \frac{1 + a_{12} a_{21}}{a_{22}} = \frac{a_{11} a_{22}}{a_{22}} = a_{11}.$
Toutes les égalités (3) seront donc satisfaites et l'égalité (1) aura lieu.
(*) Nous désignons par la même lettre z la matrice z et l'élément se trouvant à l'intersection de la seconde ligne et de la première colonne. Il n'en résulte pas la suite confusion.

La typographie ancienne méthode : chère, lente mais de grande qualité.

La typographie scientifique en images

Springer, 1966

APPLICATIONS OF THE ADDITION THEOREM
Before studying the Legendre polynomials in detail, we shall obtain several simple results on spherical harmonics in general which depend on the addition theorem.
If we remember that every spherical harmonic of degree n can be represented as
$$(20) \quad S_n(\xi) = \sum_{k=-n}^{N(n)} a_k S_{n,k}(\xi), \quad a_k = \int_{\Omega_q} S_n(\eta) S_{n,k}(\eta) d\omega_\eta$$

we get immediately from Theorem 2
Lemma 7. For every spherical harmonic of degree n
$$\frac{N(n)}{\omega_q} \int_{\Omega_q} P_n(\xi \cdot \eta) S_n(\eta) d\omega_\eta = S_n(\xi)$$

Here the letter η in connection with $d\omega_\eta$ means that the integration is carried out with respect to η .
Observing that
$$\int_{\Omega_q} S_n^2(\xi) d\omega_\xi = \sum_{k=-n}^{N(n)} (a_k)^2$$

we get from (20), using Schwarz's inequality and Theorem 2,
$$(21) \quad |S_n(\xi)|^2 \leq \sum_{k=-n}^{N(n)} (a_k)^2 \sum_{k=-n}^{N(n)} |S_{n,k}(\xi)|^2 = \frac{N(n)}{\omega_q} P_n(\xi) \sum_{k=-n}^{N(n)} (a_k)^2.$$

This gives us
Lemma 8. Let $S_n(\xi)$ be a spherical harmonic of degree n . Then
$$|S_n(\xi)| \leq \sqrt{\frac{N(n)}{\omega_q}} \int_{\Omega_q} |S_n(\xi)|^2 d\omega_\xi.$$

Springer, 1985

Thanks to Sergey Lvovich Sobolev, we now have a more potent tool for the definition of Hilbert spaces with various differentiability properties, viz. the Sobolev spaces H^s . They have gained great importance in the theory of partial differential equations, especially for existence questions where precise statements on the "regularity" of solutions are desired which frequently are impossible in the language of C^k -Banach spaces. Since Sobolev spaces are an established part of modern analysis, we may keep it short and refer for the rest to the abundant textbook literature: [Bers-John-Schechter 1964, Chapter III/IV], [Hörmander 1963, 33-63], [Lions-Magenes 1968, 1-118], [Narasimhan 1973, 184-200], [Palais 1965, 125-174], and [Yosida 1965/1974, 55 and 175 ff.].
B. Definition
In the following, we put together some of the various customary equivalent definitions of Sobolev spaces. Here, we restrict ourselves to the "case of functions" ($s \geq 0$). In the framework of distribution theory, the spaces H^s can be treated clearly and uniformly also for $s < 0$; e.g., see [Hörmander 1963], [Lions-Magenes 1968] and [Adams 1975].
Exercise 1. Show that, for $u \in C_0^\infty(\mathbb{R}^n)$, the following norms are equivalent ($n \in \mathbb{N}$):
a) $\|u\|_n := \left(\int_{|\alpha| \leq n} |D^\alpha u|^2 \right)^{1/2}$, where
$$\|u\|_0^2 := \langle u, u \rangle_0 = \int_{\mathbb{R}^n} u(x) \overline{u(x)} dx$$

and
$$D^\alpha := (-i)^{|\alpha|} \frac{\partial^{|\alpha|}}{\partial x_1^{|\alpha|}} \dots \frac{\partial^{|\alpha|}}{\partial x_n^{|\alpha|}}, \quad |\alpha| := \alpha_1 + \dots + \alpha_n.$$

b) $\|u\|_n := (2\pi)^{-n/2} (1 + |\xi|^2)^{n/2} \hat{u}|_0$
Hint: For $a \rightleftharpoons b$, start with the Fourier differentiation formula (see Chapter I.8), giving

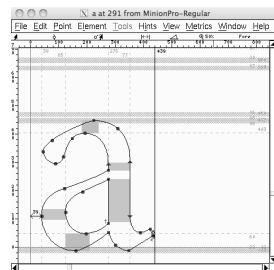
La typographie à la photocopieuse : économique, rapide mais laide.

Un peu de vocabulaire typographique (suite)

métrique d'une fonte La métrique d'une fonte est l'ensemble des informations numériques relatives à la taille, à la largeur, à l'espacement, *etc* des glyphes de la fonte.

Les logiciels de composition utilisent ces données pour placer les caractères sur la ligne d'écriture, pour rapprocher ou au contraire écarter des caractères entre eux, pour gérer la mise en page (justification, césures...), *etc*.

Exemple du caractère “a” de la fonte MINIONPRO REGULAR :



Les lignes rouges symbolisent les lignes d'écriture basse, médiane et haute...

Logiciels de composition de texte

Il existe plusieurs catégories de logiciels pour gérer du texte.

Éditeur de texte C'est un logiciel destiné à la création et à l'édition de fichiers textes sans mise en forme des polices, des couleurs, de justification...

Les fichiers textes produits sont légers et exploitables par d'autres éditeurs de texte.

→ Coloration syntaxique (FORTRAN, C, PERL, PHP, HTML, CSS... et bien sûr L^AT_EX)

→ Interface avec d'autres outils (autocomplétion, compilateurs, systèmes de contrôles de versions, archivage...).

Bloc-notes⁴⁰ (🍏), Emacs⁴¹ (🍏), Pico⁴² (🍏), TextEdit⁴³ (🍏), Fraise⁴⁴ (🍏)...

Traitement de texte C'est un logiciel qui produit une composition du texte (au sens typographique) à partir des données textuelles fournies par l'utilisateur.

Le rendu est composé en direct lors de la frappe, c'est ce qu'on appelle un logiciel WYSIWYG (*What You See Is What You Get*).

Le format de sauvegarde retient les informations de composition : mise en forme, couleurs, polices...

→ Les fichiers sont plus gros.

→ Les fichiers sont souvent moins exploitables par d'autres logiciels (pas de standard).

Microsoft Word⁴⁵ (🍏), OpenOffice⁴⁶ (🍏)...

Logiciels de composition de texte (suite)

Logiciel de mise en page (PAO³²) Dans le cadre de la “Publication Assistée par Ordinateur”, c'est un logiciel qui réalise une composition de très grande qualité typographique à partir de textes et d'images (livres, magazines, brochures...).

→ maquette, calques, chaînage de textes, habillage d'images, crénage et approche de paire, calibration des couleurs...

Il est souvent complété dans la chaîne de production par des logiciels de graphisme (photographies, dessins vectoriels...).

→ Logiciels haut de gamme, donc souvent chers.

→ Apprentissage difficile, souvent réservés à des professionnels.

InDesign⁴⁷ (🍏), QuarkXPress⁴⁸ (🍏), Scribus⁴⁹ (🍏)...

T_EX Ce n'est pas à proprement parler un “traitement de texte”, mais plutôt un “traitement de document”, au sens où il analyse un texte source pour fournir le résultat “compilé” sous forme d'un document final.

→ Rivalise avec les logiciels de mise en page grâce à un langage de programmation et à sa grande flexibilité.

→ Sa mise en œuvre est très différente des autres logiciels, ce qui dérouté les habitués des traitements de texte et des logiciels de PAO.

→ L^AT_EX se charge automatiquement de nombreux aspects et détails typographiques.

→ Sa démarche ressemble à ce qui se passe avec des navigateurs WEB : composition à partir d'un fichier texte source en HTML⁵⁰.

Une petite chronologie de l'histoire de T_EX

1978 : D. E. Knuth crée T_EX, avec son langage, son moteur de compilation tex⁵¹, et un ensemble de macros, PLAIN T_EX, regroupées sous forme d'un format.

Premier moteur tex sur 7 bits en entrée : \ ' e pour encoder é.

Rappel : en 1978, peu de “Personnal Computer”, écrans textuels, mémoires de quelques Ko...

Principe : on part d'un fichier source en texte brut (.tex)

→ “compilation” à l'aide du moteur tex et des macros de PLAIN T_EX,

→ fichier de description de la page en DVI⁵² (.dvi, “device independant”),

→ impression (drivers divers : dvips pour une sortie en POSTSCRIPT⁵³ par exemple).

1982 : L. Lamport introduit L^AT_EX⁵⁴, un autre jeu de macros au dessus de T_EX, regroupées sous forme d'un format, plus simple à utiliser que PLAIN T_EX.

C'est surtout un langage de *description sémantique* du texte.

→ la compilation utilise toujours le moteur tex.

Apparition des *packages* (“modules” en français)

→ extension facilitée des fonctionnalités.

1989 : La version 3 du moteur tex permet de gérer des caractères sur 8 bits (256 caractères différents), donc de lire des textes avec des lettres accentuées.

1994 : L^AT_EX2e remplace L^AT_EX2.09 qui vieillissait mal, en particulier par l'anarchie dans les noms des *packages* et leur incompatibilité.

Une petite chronologie de l'histoire de T_EX (suite)

fin des années 1990 : Hàn Thê Thành introduit le moteur `pdftex` (son travail de thèse) : sortie par défaut dans le format PDF³⁴ (*portable document format* inventé par Adobe), gestion des polices vectorielles, extensions micro-typographiques, accès à des fonctionnalités PDF (hyperliens, table des matières...).

Aujourd'hui c'est le moteur par défaut dans les installations de T_EX.

2008 : Le moteur `tex` atteint la version 3.1415926 (corrections de bugs).

Aucune fonctionnalité n'est ajoutée à `tex` depuis la version 3.

Chaque correction de bug ajoute une décimale de π ⁵⁴.

2008 : Première version publique du moteur `xetex`.

Extension de `pdftex` pour utiliser les polices installées sur le système d'exploitation, codage en entrée UNICODE⁵⁵ (16 bits).

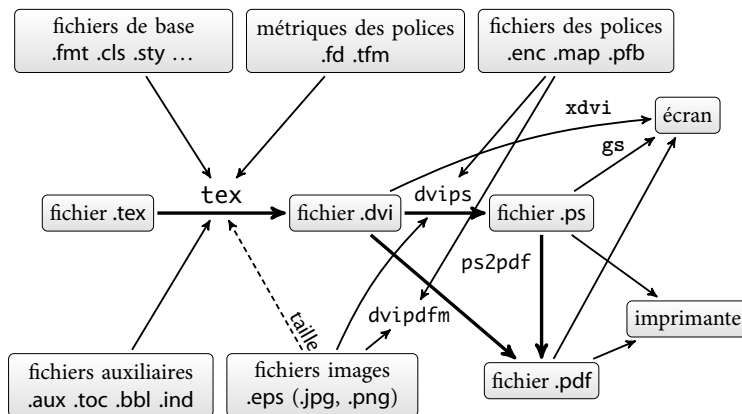
2010 : Première version publique du moteur `luatex`⁵⁶.

Fusion du meilleur de `pdftex` et de `xetex`, ouverture de la composition des pages au langage de programmation LUA⁵⁷.

futur proche (5 ans) : `xetex` et `luatex` vont remplacer `pdftex` : utilisation des dernières technologies en matière de polices vectorielles (TRUE_{TYPE}⁵⁸, OPEN_{TYPE}⁵⁹), en particulier le "standard" développé par Microsoft et Adobe sur les polices mathématiques.

futur toujours trop loin : Le projet L^AT_EX3 doit remplacer L^AT_EX2_ε. 20 ans qu'on attend...

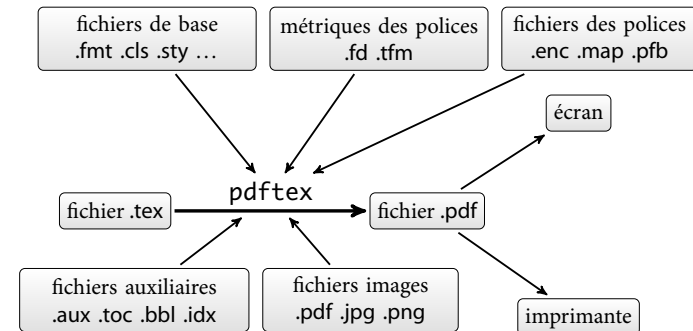
Flux de production de L^AT_EX avec le moteur `tex`



La complexité de ce diagramme explique pourquoi L^AT_EX n'est pas simple à maîtriser...

La commande `latex` correspond en fait à lancer `tex` avec le format L^AT_EX (latex.fmt).

Flux de production de L^AT_EX avec le moteur `pdftex`



Flux équivalent avec les moteurs `xetex` et `luatex` avec en plus l'accès aux polices du type .tff (TRUE_{TYPE}⁵⁸) ou .otf (OPEN_{TYPE}⁵⁹) installées sur le système d'exploitation hôte.

Dans les installations récentes de T_EX, c'est le moteur `pdftex` qui est utilisé, même dans la chaîne de production passant par un fichier DVI.

⚠ Ça ne signifie pas que les deux chaînes de production soient les mêmes !

Les avantages de L^AT_EX

- Le fichier source est du texte.
 - ➔ Taille très petite : quelques Mo pour un livre de 600 pages.
 - ➔ Très grande portabilité (tous les systèmes d'exploitation).
 - ➔ Le texte peut être généré par un logiciel tiers (insertion dans un flux automatisé).
- Logiciels gratuits, ouverts et stabilisés ➔ pérennité des documents.
- Typographie de très grande qualité due au moteur `tex` : césures, ligatures...
- Possibilité de programmation : macros personnelles, mise en page, aspect des éléments du texte, automatisations diverses...
- Séparation du fond et de la forme.
 - ➔ Changement de style aisé.
 - ➔ Gestion automatique de nombreux éléments du document (table des matières, références croisées, bibliographie...).
- Capacité à gérer des gros documents complexes.
- Gestion aisée de documents écrits dans des langues et dialectes différents.
 - ➔ très utilisé en linguistique.
- Nombreuses extensions sous forme de *packages*.
- Écosystème riche : `makeindex`, `bibtex`, `metapost`...
- *Last but not least* : excellente composition des formules mathématiques !

Les inconvénients de L^AT_EX

- Installation et maintenance peu aisées, mais ça s'améliore.
Depuis TeXLive 2009 on peut mettre à jour les *packages* par l'utilitaire tlmgr.
 - Non WYSIWYG (*What You See Is What You Get*).
Le document de départ est loin visuellement du document final.
Les environnements de travail d'aujourd'hui pallient largement à ce problème.
 - Extrême rigueur requise pour la composition (en fait, c'est un avantage).
 - Limitations typographiques sur certains points précis (forme des paragraphes...) où la concurrence WYSIWYG fait largement mieux (InDesign¹¹, QuarkXPress⁴⁸).
 - Il n'y en a guère des polices ! Double origine :
 - T_EX fonctionne avec ses propres polices, dans un format très spécifique.
 - Peu de polices complètes existent pour les mathématiques.
- xetex et luatex remédient à ces problèmes en accédant aux polices installées sur le système d'exploitation hôte et en accédant à des polices mathématiques nouvelles (voir aussi le projet STIX⁶⁰).
- Le langage de programmation de T_EX est différent des langages de programmation usuels : il est difficile à maîtriser et très déroutant (consulter les forums !).
L'ouverture au langage LUA⁵⁷ est peut-être une solution.
 - Limitations intrinsèques dues à une programmation ancienne (ordinateurs avec peu de ressources) sur le nombres de fontes en mathématiques, le nombre de registres...

1.3 L'installation de T_EX et son écosystème

L'installation de T_EX

Pour utiliser L^AT_EX il faut installer deux types de ressources :

1. Des logiciels ("moteurs") tex, pdftex, bibtex, makeindex... et des dizaines de milliers de fichiers regroupés à différents endroits sur la machine (classes, *packages*, polices...). Il existe pour ça des distributions prêtes à l'emploi :
TeXLive¹¹ (🍏🍌), MacTeX¹⁸ (TeXLive pour 🍏), proTeXt²⁴ (TeXLive pour 🍌).

2. Une interface conviviale qui contient :
 - un éditeur de texte pour écrire le fichier source ;
 - un moyen de lancer la compilation (terminal intégré, menu ou bouton) ;
 - un logiciel (intégré ou non) de visualisation du DVI, du PS ou du PDF produit ;
 - des fonctionnalités diverses qui permettent une meilleure productivité : aide à la frappe (auto-complétion), correction orthographique, aller-retour "source"/"résultat visuel", interface avec d'autres moteurs...

Il existe pour ça des logiciels intégrés :

TeXShop¹⁹ (🍏), TeXworks¹² (🍏🍌), TeXMaker¹³ (🍏🍌), emacs¹⁴ (🍌), Aquamacs²⁰ (🍏), TeXnicCenter²⁶ (🍌), WinEdit²⁷ (🍌)...

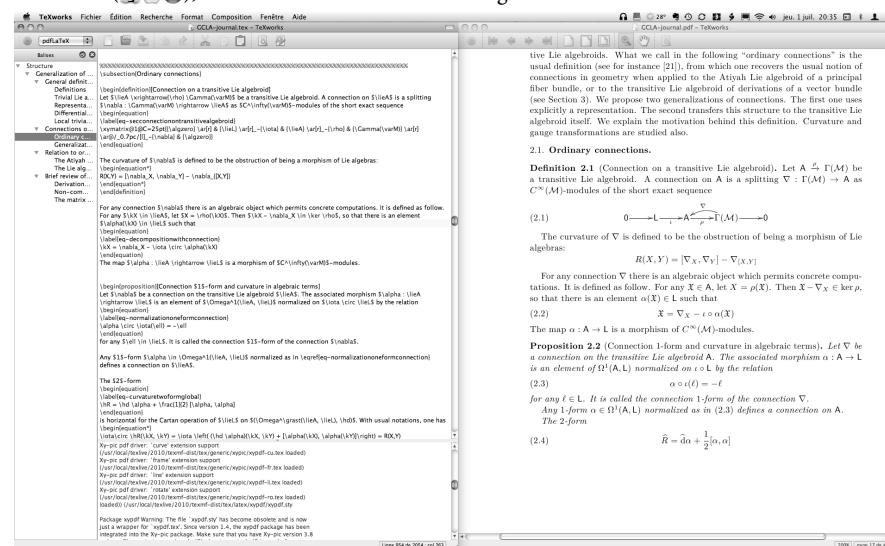
Ma préférence va à TeXShop pour son interface sobre, conviviale et efficace.

Son clone multiplateforme TeXworks reprend tous ces avantages.

On peut aussi installer un gestionnaire de bibliographie compatible avec bibtex, un logiciel de gestion des *packages* compatible avec tlmgr...

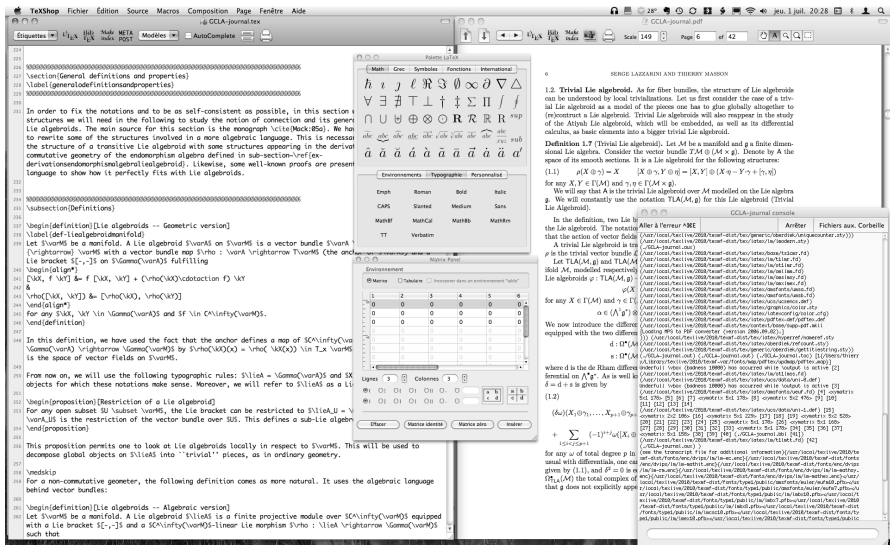
Un exemple d'environnement de travail

TeXworks (🍏🍌), un environnement de travail intégré :



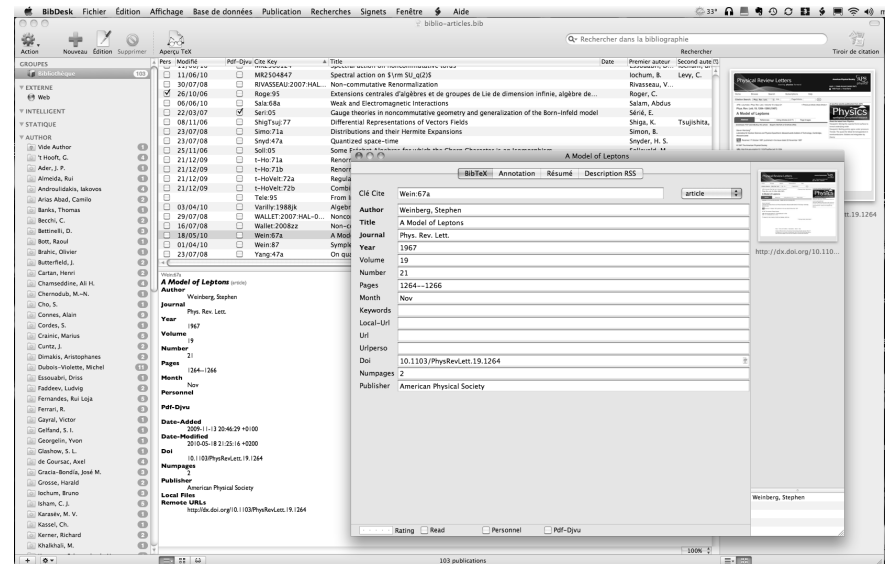
Un exemple d'environnement de travail (suite)

TeXShop (🍏), un environnement de travail intégré :



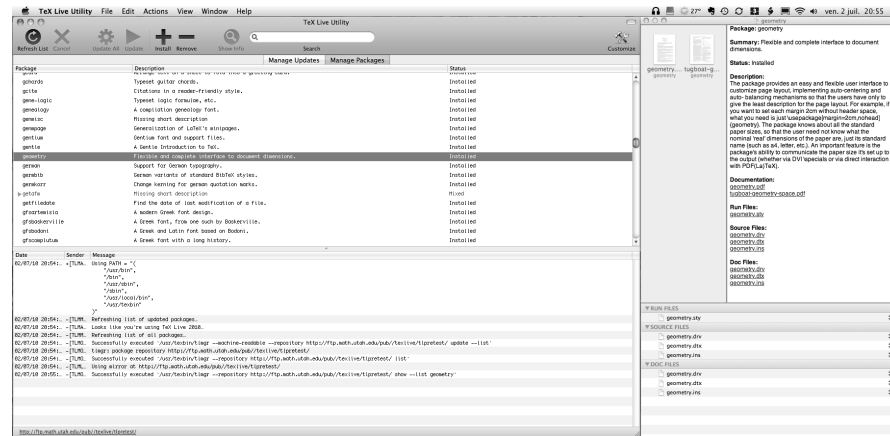
Un exemple d'environnement de travail (suite)

BibDesk (🍏), gestionnaire de fichiers de bibliographie pour bibtex :



Un exemple d'environnement de travail (suite)

TeX Live Utility (🍏), interface de tlmgr :



Quelques logiciels annexes

Autour du moteur tex, on trouve de nombreux autres logiciels très utiles :

dvi [qqchose] Divers utilitaires pour transformer un fichier DVI⁵² dans d'autres formats :
 dvips pour obtenir du PS⁵³,
 dvi2pdf pour obtenir du PDF⁵⁴,
 dvi2png pour le format d'image PNG⁵⁶,
 etc.

ps[qqchose] Divers utilitaires pour manipuler un fichier PS :
 ps2eps pour obtenir un fichier EPS⁶² (il détermine la "BoundingBox"),
 pstotopdf pour obtenir un PDF,
 pstops pour nettoyer/corriger un PS,
 psnup, psbook, etc.

texcount Logiciel qui compte des mots dans des fichiers T_EX ou L_AT_EX en ignorant les macros, les tableaux, les formules...

Latexdiff Logiciel qui compare deux fichiers L_AT_EX et qui crée une version commune avec commentaire des différences.

Quelques logiciels annexes (suite)

tlmgr (*T_EX Live Manager*) Cet utilitaire permet de gérer les *packages* installés sur l'ordinateur et de les actualiser en les téléchargeant sur le site du CTAN⁵ (**Comprehensive T_EX Archive Network**).

Idéal pour maintenir son installation T_EX à jour.

Il existe des interfaces graphiques pour l'utiliser (**TeX Live Utility**...).

La commande la plus utile qu'il faut invoquer en tant qu'administrateur est

```
tlmgr update --all
```

→ elle met à jour tous les *packages* et les logiciels.

bibtex Logiciel d'extraction et de mise en forme de données bibliographiques.

Il permet de conserver toutes les données bibliographiques dans un fichier .bib, et d'utiliser ces données dans n'importe quel projet.

Le style de mise en forme est géré par des fichiers .bst.

makeindex Logiciel qui trie les données d'index d'un fichier .idx produit par L^AT_EX et les met en forme.

metapost, **asymptote**⁶³ Logiciels de dessin basé sur le même principe que T_EX : document source "texte" → .eps ou .pdf.

Avantages : langage de programmation pour décrire le dessin, insertion de code T_EX.

L'utilisation de **bibtex**, **makeindex** et **asymptote** feront l'objet d'autres cours.

Les dossiers texmf

Description d'une distribution **TeXLive** récente (2010).

Les fichiers d'une installation T_EX sont regroupés et structurés dans des dossiers **texmf**.

Il y a plusieurs dossiers **texmf** :

- L'installation principale et fondamentale se trouve dans un ensemble de dossiers **texmf** auxquels personne n'est censé toucher...

Typiquement, ils sont dans `/usr/local/texlive/2010/` et portent des noms comme : **texmf**, **texmf-config**, **texmf-dist**, **texmf-var**.

- L'administrateur de la machine peut installer des fichiers complémentaires à l'installation fondamentale dans un dossier **texmf-local** :

`/usr/local/texlive/texmf-local/`

Tout apport dans ce dossier est utilisable par tous les utilisateurs de la machine.

- Chaque utilisateur dispose d'un **texmf** personnel :

`$HOME/texmf/` (🏠), `$HOME/Library/texmf/` (🍏),

`%USERPROFILE%\texmf\` (👤).

L'utilisateur peut y déposer les fichiers qu'il veut, il est le seul à pouvoir les utiliser.

Il y a une distribution **TeXLive** proposée tous les ans.

Le dossier **texmf-local** n'est pas touché par ces mises à jours.

Plusieurs distributions **TeXLive** peuvent cohabiter (années différentes).

Recherche de fichiers dans les dossiers texmf

Règle fondamentale

Par principe, le moteur **tex** cherche en priorité un fichier souhaité (.tex, .sty, .aux, .def, .pdf, .jpg...) dans l'ordre suivant :

1. dans le dossier courant de travail (celui du fichier source) ;
2. puis dans le dossier **texmf** de l'utilisateur ;
3. puis dans **texmf-local** ;
4. enfin dans les **texmf** fondamentaux.

Par conséquent, tout fichier dans le **texmf** de l'utilisateur est prioritaire.

On trouvera dans le fichier .log la liste des fichiers trouvés et leur chemin.

Pour gagner du temps lors de la recherche d'un fichier dans ces immenses dossiers, chaque dossier **texmf** comporte un fichier **ls-R**, qui contient toute la liste hiérarchique des fichiers qu'il renferme.

→ lorsqu'on ajoute un fichier dans un tel dossier, il faut actualiser le fichier **ls-R**.

C'est facile, il faut et il suffit de taper la commande **mktexlsr** dans un terminal.

⚠ Pour que les fichiers déposés dans votre dossier **texmf** soient pris en compte, pensez à taper la commande **mktexlsr** (pas nécessaire sous **MACOSX**).

Structure des dossiers texmf

Chaque dossier **texmf** est structuré en sous-dossiers. Parmi ceux-ci on trouve :

bibtex Ce dossier contient lui-même deux sous-dossiers intéressants :

bib pour les fichiers de bibliographie .bib

bst pour les fichiers de style bibliographiques .bst.

doc Ce dossier contient la documentation sur l'installation, les logiciels et les *packages*.

fonts Ce dossier contient tout ce qui concerne les polices d'écriture, organisés dans un fouillis (ordonné !) de sous-dossiers...

tex C'est le dossier le plus intéressant. Il contient lui-même d'autres sous-dossiers, dont :

generic Pour les fichiers et dossiers consacrés à tous les formats.

latex Pour les fichiers et dossiers consacrés au format L^AT_EX.

web2c Ce dossier contient certains fichiers de configuration de l'installation T_EX et les formats .fmt.

Il existe de nombreux autres dossiers et sous-dossiers, mais ils n'intéressent pas l'utilisateur usuel de L^AT_EX.

En particulier chaque moteur a son propre dossier : **asymptote**, **dvipdfmx**, **dvips**, **makeindex**, **metapost**, **xdvi**...

Le dossier texmf de l'utilisateur

Le dossier `texmf` de l'utilisateur peut servir à :

- placer un *package* absent de l'installation par défaut ou qui n'est pas à jour ;
- placer des fichiers de style personnels ou récupérés sur l'Internet (`.sty`, `.bst`...), par exemple des feuilles de style ou des classes de revues où on publie ;
- organiser des fichiers communs à différents projets : des images (logos d'institut par exemple), des macros regroupées dans un fichier `.tex` puis appelé dans le document...
- placer les bases de données bibliographiques qu'on gère (`.bib`).

Ce dossier `texmf` doit être structuré de la même façon que les autres dossiers `texmf`, mais sans nécessairement contenir autant de dossiers.

On ne crée donc un dossier/sous dossier que lorsque c'est nécessaire.

- Les fichiers des *packages* `.sty` iront dans `texmf/tex/latex/`.
- Les bases de données bibliographiques `.bib` iront dans `texmf/bibtex/bib/`.
- Les fichiers de style bibliographiques `.bst` iront dans `texmf/bibtex/bst/`.
- Les fichiers d'images `.eps`, `.pdf`, `.jpg`, `.png`... iront dans `texmf/tex/latex/`.

On peut créer des sous dossiers sans contrainte au delà de ces dossiers. Par exemple il est souhaitable de créer un dossier images dans `texmf/tex/latex/` pour regrouper les images et de regrouper dans des dossiers spécifiques les styles des revues...

L'installation d'un *package* dans son `texmf`

Les étapes à suivre pour installer un *package* dans son propre dossier `texmf` :

1. Récupérer les fichiers du *package*.
La **page de recherche** du CTAN⁶⁴ permet de localiser un *package* en cherchant le nom d'un fichier précisément ou en cherchant des mots clés dans sa description (*Search the package descriptions*, le plus efficace).
Une fois trouvé le *package* à une URL du type
`http://tug.ctan.org/tex-archive/macros/latex/contrib/geometry/`
on peut obtenir le *package* sous forme d'un dossier archive (`.zip`).
2. Décompresser ce fichier et placer le dossier obtenu dans votre `texmf/tex/latex/` (ou autre selon la nature du *package*).
3. Dans ce dossier doit figurer un fichier de type `.ins` et/ou `.dtx`.
Lancer la compilation de `pdflatex` sur `.ins` (de préférence) ou sur `.dtx`.
Pour ça on peut passer par un terminal : `pdflatex geometry.ins`
ou on peut ouvrir ce fichier dans son éditeur T_EX préféré...
Cette compilation produit les fichiers du *package* (`.sty`, `.fd`, `.cfg`, `.def`...).
Si le dossier ne contient pas un tel fichier, il n'y a rien à faire à cette étape.
4. Lancer la commande `mktexlsr` (pas nécessaire sous MACOSX).

Dans l'absolu, si on en a la possibilité, il est préférable de mettre à jour les *packages* avec l'utilitaire `tlmgr` ou mieux encore avec une interface graphique à cet utilitaire.

Les types de fichiers de l'univers L^AT_EX

Fichiers visibles côté utilisateur, essentiellement dans le dossier de travail :

Fichiers créés par l'utilisateur

- `.tex` Fichier source T_EX ou L^AT_EX qui est compilé par le moteur `tex`.
- `.bib` Base de donnée bibliographique, utilisée par `bibtex`.

Fichiers générés par L^AT_EX

- `.log` Fichier donnant les détails de ce qui s'est passé lors de la compilation.
- `.aux` Informations sur le documents, utilisé par une seconde compilation.
- `.toc` Informations pour la table des matières.
- `.lof` Informations pour la liste des figures.
- `.lot` Informations pour la liste des tableaux.
- `.idx` Informations pour créer un index.
- `.dvi` Fichier de sortie produit par le moteur `tex`.
- `.pdf` Fichier de sortie produit par le moteur `pdftex`.

Fichiers générés par d'autres moteurs

- `.ind` Fichier généré à partir de `.idx` par `makeindex`.
- `.ilg` Fichier log du moteur `makeindex`.
- `.bbl` Fichier généré à partir de `.aux` et `.bib` par `bibtex`.
- `.blg` Fichier log du moteur `bibtex`.

Les types de fichiers de l'univers L^AT_EX (suite)

Fichiers plus fondamentaux où l'utilisateur usuel n'intervient guère :

Fichiers utilisés par L^AT_EX

- `.cls` Fichier de "classe" appelé par `\documentclass[-]{-}`.
- `.clo` Fichier d'option de "classe" appelé par `\documentclass[-]{-}`.
- `.sty` Fichier de "style" (*packages*) appelé par `\usepackage[-]{-}`.
- `.cfg` Fichier de configuration de certains *packages* (**graphics**, **hyperref**...).
- `.def` Fichier de définition associés à certains *packages* (**inputenc**, **babel**...).
- `.fd` Fichier de description des polices d'écriture.
- `.vf` Fichier de polices virtuelles.
- `.tfm` Fichier de métrique des fontes.
- `.dtx` Fichier de distribution des *packages* (produit les `.sty`, `.def`, `.pdf`...).
- `.ins` Fichier gérant l'installation d'un *package* par une compilation L^AT_EX.

Fichiers utilisés par d'autres moteurs

- `.map` Fichier de "mappage" entre les noms des polices (`pdftex`, `dvips`...).
- `.enc` Fichier d'encodage des polices (`pdftex`, `dvips`...).
- `.pfb` Fichier des polices vectorielles (`pdftex`, `dvips`...).
- `.bst` Fichier de style bibliographique (`bibtex`).
- `.ist` Fichier de style d'index (`makeindex`).

Les types de fichiers de l'univers L^AT_EX (suite)

Remarque 1 : Des classes et des *packages* peuvent générer leur propres fichiers auxiliaires avec d'autres extensions dans le dossier de travail.

La liste donnée ci-dessus est donc non exhaustive.

Par exemple **ntheorem** produit un fichier auxiliaire `.thm` pour y placer les métadonnées relatives aux environnements de type "théorème" qu'il gère.

Remarque 2 : Les fichiers de polices d'écriture sont très nombreux dans une installation T_EX et sont de divers types : `.afm`, `.enc`, `.fd`, `.lig`, `.map`, `.mf`, `.otf`, `.pfb`, `.pk`, `.tfm`, `.ttf`, `.vf`...

C'est l'un des aspects les plus compliqués d'une installation T_EX...

Remarque 3 : Chaque *package* est fourni avec des fichiers `.ins` et `.dtx` : la compilation du fichier `.ins` crée et installe les fichiers du *package*.

À la compilation, le fichier `.dtx` produit la documentation et le `.sty` du *package* (voire d'autres fichiers).

Remarque 4 : Le premier fichier utilisé par le moteur `tex` pour compiler en L^AT_EX est `latex.fmt` qui définit L^AT_EX en tant que format (`.fmt`).

C'est une version pré-compilée d'un fichier nommé `latex.ltx` (8 000 lignes de code) dans lequel figurent toutes les macros constituant L^AT_EX.

Ce dernier fichier est consultable pour comprendre comment sont définies certaines commandes ou certains environnements.

Remarque 5 : Les autres logiciels fournis avec le moteur `tex` ont aussi à leur disposition des fichiers particuliers, avec leurs propres extensions.

À la recherche d'une documentation

Pour tout ce qui concerne les commandes et les environnements définis par L^AT_EX, consulter **L^AT_EX Reference Manual**^{⌚4} (versions HTML ou PDF disponibles).

Les logiciels et les *packages* installés sur l'ordinateur sont en général documentés.

On peut facilement accéder à cette documentation avec la commande `texdoc` :

```
texdoc geometry
```

ouvre le PDF de la documentation du *package* **geometry**.

→ fonctionne pour à peu près tous les *packages*.

```
texdoc article
```

ouvre le PDF de documentation des classes standard de L^AT_EX.

```
texdoc pdftex
```

ouvre le PDF de documentation de `pdftex`.

```
texdoc texlive
```

ouvre le PDF de documentation de T_EXLive.

```
texdoc lshort et texdoc symbols
```

ouvrent les PDF de **The Not So Short Introduction to L^AT_EX**^{⌚2} et

Comprehensive L^AT_EX symbol list^{⌚3} respectivement.

Pour plus de détails et d'exemples, consulter des livres et des cours/fascicules.

À la recherche d'un fichier

En interne, tous les logiciels de l'installation T_EX délèguent à un utilitaire le soin de trouver un fichier requis dans tous les dossiers `texmf`.

Il existe une commande associée à cet utilitaire : `kpsewhich`.

Par exemple, pour chercher le fichier `textcomp.sty`, dans un terminal on tape :

```
kpsewhich textcomp.sty
```

Cette commande retourne :

```
/usr/local/texlive/2010/texmf-dist/tex/latex/base/textcomp.sty
```

La recherche procède comme indiquée plus haut dans l'ordre des priorités :

dossier courant → `texmf` utilisateur → `texmf-local` → `texmf` fondamentaux

Le résultat est le premier chemin trouvé vers le fichier (ou rien si le fichier est inconnu).

La commande fonctionne avec presque tous les types de fichiers, le nom doit être exact.

```
kpsewhich -all textcomp.sty
```

retourne toutes les occurrences trouvées.

Cours 2 – Les fondamentaux : le document source

« Mon pauvre oncle disait souvent :
"Il faut toujours tourner sa langue sept fois
dans sa bouche avant de parler."
Que devrait-on faire avant d'écrire ? »
Gérard de Nerval
Les illuminés

2.1 Le document source : les bases	24
Où l'on prend contact avec les rudiments d'un langage cabalistique grâce auquel des incantations magiques feront de nous les maîtres de la page blanche...	
2.2 Structure du document source	29
Où l'on découvre enfin le vrai visage d'un document source, et où l'on s'instruit de la diversité des classes et des modules...	
2.3 Repères sur les concepts de programmation	35
Où l'on commence à découvrir la grammaire parfois obscure de notre langage cabalistique...	
2.4 Tour d'horizon de commandes de base	39
Où l'on se donne un peu de vocabulaire pour disposer à notre guise de la page blanche...	

2.1 Le document source : les bases

La philosophie de T_EX

Avec T_EX et L^AT_EX, on tape du texte (presque) normalement et on fait appel de temps en temps à des macros (des morceaux de programmes) pour obtenir des effets divers ou des fonctionnalités particulières.

△ Il y a des caractères réservés pour la syntaxe des macros.

En voici la liste, ainsi que la façon de les obtenir concrètement dans du texte :

<code>~ \~{ }</code>	<code>\$ \\$</code>	<code>& \&</code>	<code>% \%</code>	<code># \#</code>
<code>^ \^ { }</code>	<code>_ _</code>	<code>{ \{ }</code>	<code>} \}</code>	<code>\ \textbackslash</code>

(voir aussi le *package* **textcomp**)

Avec des macros on peut : modifier les styles typographiques (taille, graisse, forme, couleur...), modifier la mise en page (marges, indentations...), ajouter des notes en bas de page, composer des titres, générer une table des matières, composer des mathématiques (symboles, disposition des formules...), insérer des images, créer des tableaux, composer des dessins...

Le fait que T_EX soit avant tout un langage de programmation permet de créer des macros capables de remplir de très nombreuses tâches.

Parmi ces macros, figurent les “commandes”, sous la forme `\cmd` (“\” suivi d’un nom).

T_EX ne connaît que des commandes avec ou sans arguments.

L^AT_EX a introduit les arguments optionnels et la notion d’environnement.

Le rôle des caractères réservés

Les caractères réservés se rencontrent dans les situations suivantes :

- `\` marque le début d’une macro.
Exemple : `\newpage` est la commande pour démarrer une nouvelle page.
- `{` et `}` ouvrent et ferment un groupe, c’est à dire une entité unique pour T_EX.
Cela sert à créer un contexte local ou à englober l’argument (long) d’une macro.
Exemples : `{\large texte en grand}`, `\textit{texte en italique}`.
→ les (re)définitions opérées à l’intérieur d’un groupe restent locales.
- `%` marque le début d’un commentaire.
- `~` crée un espace insécable.
- `$` commence et termine le mode mathématique.
- Dans le mode mathématique, `^` sert à placer des exposants et `_` sert à placer des indices.
- `&` intervient dans les séparations des colonnes de tableaux et de matrices, et comme positions d’alignement de formules de mathématique.
- `#` se rencontre dans la définition de macros.

△ Les caractères `[` et `]` ne sont pas réservés, mais ils interviennent pour délimiter des arguments optionnels dans des macros L^AT_EX, ce qui fait que leur rôle est ambigu.

→ cette situation génère parfois des erreurs incompréhensibles.

Commandes et environnements, généralités

Il y a 4 types de commandes :

1. `\cmd` : sans aucun argument ;
2. `\cmd{-}` : un ou plusieurs arguments obligatoires, chacun entre accolades `{-}` ;
3. `\cmd[-]` : un ou plusieurs arguments optionnels, chacun entre crochets `[-]` ;
4. `\cmd[-]{-}` : un ou plusieurs arguments optionnels et un ou plusieurs arguments obligatoires.

Les environnements sont des macros qui englobent un morceau de texte.

De ce fait, un environnement travaille toujours avec du contenu : le texte englobé.

Comme pour les commandes, il y a 4 types d’environnements :

1. `\begin{env} ... \end{env}` : sans argument ;
2. `\begin{env}{-} ... \end{env}` : un ou plusieurs arguments obligatoires ;
3. `\begin{env}[-] ... \end{env}` : un ou plusieurs arguments optionnels ;
4. `\begin{env}[-]{-} ... \end{env}` : un ou plusieurs arguments optionnels et un ou plusieurs arguments obligatoires.

La place des arguments optionnels est souvent avant les arguments obligatoires, mais pas forcément.

Commandes et environnements, exemples

Exemples de commandes :

```
\documentclass[-]{-}, \usepackage[-]{-}
\chapter[-]{-}, \section[-]{-}, \footnote{-}, \tableofcontents
\Large, \tiny, \textbf{-}, \slshape, \textcolor[-]{-}
\par, \bigskip, \medskip, \smallskip, \vspace{-}
\newpage, \clearpage, \pagebreak[-]
\alpha, \otimes, \sum, \sin, \sqrt{-}, \frac{-}{-}
\includegraphics[-]{-}, \scalebox[-]{-}
\{, \$, \,, \backslash
```

Exemples d’environnements :

```
\begin{document} ... \end{document}
\begin{enumerate} ... \end{enumerate}
\begin{abstract} ... \end{abstract}
\begin{flushleft} ... \end{flushleft}
\begin{tabular}{-} ... \end{tabular}
\begin{minipage}[-]{-} ... \end{minipage}
\begin{equation} ... \end{equation}
\begin{figure}[-] ... \end{figure}
\begin{thebibliography}[-] ... \end{thebibliography}
```

.....

Commandes et environnements, subtilités

Certaines commandes et certains environnements semblent équivalents :

`\begin{center} ... \end{center}`, `{\centering ...}`, `\centerline{-}`.

En réalité, il y a des différences : l'environnement `center` ajoute des espaces avant et après le texte centré contrairement à `\centering`, et ces deux macros peuvent contenir des lignes interrompues par `\\` alors que `\centerline{-}` ne le peut pas.

(`\centerline{-}` est du pur \TeX , qu'il faut bannir en \LaTeX .)

Certaines commandes ne sont définies que localement.

`\item` n'est défini que dans les environnements `itemize`, `description` et `enumerate`.

Les commandes mathématiques ne sont définies que dans le mode mathématique : `$... $` ou dans les environnements `equation`, `align`, `gather`, `multline`...

Certaines commandes représentent des morceaux de texte ou des longueurs : ce sont donc plus des variables au sens de la programmation que des macros.

Par contre, les “variables” compteurs ne sont pas des commandes mais juste des noms.

⚠ Risques fréquents de confusions...

Il est facile de définir soi-même des commandes et des environnements, \LaTeX fournit les outils (les commandes !) pour ça.

Remarque technique (mais parfois utile).

Un environnement de type `\begin{env} ... \end{env}` est en réalité un ensemble de deux commandes :

`\env` (avec d'éventuels arguments et options) qui ouvre et `\endenv` qui ferme.

.....

Le B.A.BA de la frappe sous \LaTeX

- \TeX ignore ce qui est sur la même ligne au delà d'un caractère %.
➔ permet de placer des commentaires dans le fichier source et de désactiver des lignes de texte.
⚠ Le “retour chariot” à la fin d'une ligne contenant un % est aussi ignoré par \TeX .
- \TeX ignore les retours chariot isolés (un seul passage à la ligne).
➔ on peut découper un paragraphe dans le code source en lignes pas trop longues.
Typographie : un retour chariot isolé est considéré comme un espace, donc on reste dans le paragraphe courant.
- \TeX traite les tabulations comme des espaces.
➔ on peut utiliser des tabulations pour structurer le fichier source.
Typographie : il y a un environnement spécifique pour créer des tabulations.
- Pour \TeX , un ou plusieurs espaces entre mots signifie la même chose.
➔ rappel : une tabulation et un retour chariot (isolé) sont aussi des espaces !
Typographie : pas “d'espaces” superflus entre les mots, si on veut augmenter l'espace entre deux mots, il faut utiliser une commande, par exemple `\hspace{<dim>}`.
- \TeX a tendance à absorber les espaces qui suivent une commande :
`\LaTeX est formidable` ➔ \LaTeX est formidable,
`\LaTeX{}` est formidable ➔ \LaTeX est formidable.

.....

Le B.A.BA de la frappe sous \LaTeX (suite)

- `\\` interrompt la ligne courante et passe à la ligne.
➔ à utiliser surtout avec des environnements `center`, `flushleft`, `flushright`...
Typographie : la nouvelle ligne ne commence pas un nouveau paragraphe.
`\\[<dim>]` ajoute en plus la longueur `<dim>` verticalement entre les lignes.
`*[<dim>]` interdit en plus un saut de page.
➔ syntaxes valables aussi bien dans le texte qu'en mathématique.
- Une ligne vide (ou plus) signifie “passage dans un nouveau paragraphe”.
➔ à user sans modération pour aérer le code source.
Typographie : arrêt du paragraphe courant, passage à la ligne avec éventuel espace vertical supplémentaire, indentation de la nouvelle ligne...
Equivalut à placer `\par` à la fin de la ligne du paragraphe à terminer.
- Un `_` (\backslash suivi d'un espace) créé un espace de largeur fixe.
Un `~` collant deux mots crée un espace de largeur fixe et insécable.
Typographie : les espaces habituels entre les mots ont une largeur variable pour permettre la justification des lignes.
- Du code encadré par le signe `$` correspond au mode mathématique dans le texte.
➔ c'est ce qu'on appelle des mathématiques *textstyle*.
Il y a le mode *displaystyle* qui place les formules hors des paragraphes.
Typographie : la forme (*shape*) passe à “italique”, les espacements entre lettres sont constants, certaines macros sont activées (`_` et `^`), les accents de texte sont interdits...

.....

Conseils élémentaires pour maîtriser \LaTeX

Apprendre \LaTeX , c'est apprendre le langage \TeX/\LaTeX ainsi qu'un logiciel d'interface.

Règle 1 : Si votre clavier n'est pas américain, *trouver comment produire le symbole* `\`.

Règle 2 : Rechercher un symbole inconnu dans **Comprehensive \LaTeX symbol list** ^③.

Règle 3 : Rechercher la syntaxe d'une macro dans **\LaTeX Reference Manual** ^④.

L'interface : C'est le lieu de travail, donc l'essentiel.

- Choisir un environnement de travail riche, puissant et surtout bien maîtrisé.
- Créer un ensemble de fichiers d'exemples (vides de contenu sémantique) pour différents type de documents : articles, lettres, mémoires, rapports, CV...
Certains logiciels gèrent de tels modèles qu'il est facile d'appeler lors de la création d'un nouveau document.

Gestion du code source : Le code source, c'est le reflet de notre personnalité...

- Aérer le texte source, mettre des lignes de commentaires et des indentations.
Pour \LaTeX , une ou plusieurs lignes vides, c'est la même chose.
- Marquer clairement le début et la fin des groupes (`{ ... }` ou `\begin{env} ... \end{env}`).
- S'il s'agit d'un gros projet (rapport, livre), découper le texte en petits morceaux, et définir un fichier “maître” qui appelle les autres avec des `\include{-}`.
➔ Chaque fichier de travail (chapitre par exemple) est plus petit.
➔ Commenter certains `\include{-}` pour ne pas compiler tout le document.

Conseils élémentaires pour maîtriser L^AT_EX (suite)

Gestion des erreurs : Les erreurs sont inévitables, il faut savoir les gérer.

- À l'ouverture d'un groupe, le fermer immédiatement avant de le remplir.
- Compiler souvent le texte pour éviter d'accumuler des erreurs et pour pouvoir mieux cerner les zones en cause.
 ⚠ Une erreur à un endroit peut se révéler dans la compilation bien plus loin !
- Observer la console de compilation, et si ça ne suffit pas, ouvrir le fichier .log.
 C'est une mine d'informations qui permet de suivre le déroulement de la compilation (plus d'infos dans le .log que dans la console) et de comprendre la source des erreurs.
- En cas de problème, déplacer le `\end{document}` vers le haut jusqu'à trouver la zone qui ne compile pas ou mettre en commentaires de grands blocs de texte (certains logiciels le permettent facilement).

Personnalisation : Ne pas hésiter à personnaliser votre façon de travailler.

- Définir des macros sémantiques, qui auront un sens, et dont il sera possible de modifier le comportement jusqu'à la dernière compilation.
`\section{-}` est une macro sémantique qu'il est possible de modifier.
 Dans cette présentation : `\dossier{texmf}` produit `texmf` et `\pkge{geometry}` produit `geometry`...
- Ne pas hésiter à utiliser un *package* qui fera le travail demandé...

De l'auteur à l'imprimeur

En tant qu'auteurs, nous sommes les premiers maillons d'une chaîne de production qui se destine en général à produire un document imprimé.

Où s'arrête notre activité dans cette chaîne ?

Il y a en gros deux types de situations.

1. L'auteur rend sa copie à un éditeur qui finalise la production.
 ➔ article de recherche, chapitre d'un livre collectif...
2. L'auteur contrôle jusqu'au bout l'aspect final du document.
 ➔ cours distribué au format PDF, rapport d'activité...

Dans le premier cas, l'auteur doit fournir le code source L^AT_EX :

- Ce code source doit rester le plus portable possible.
 ➔ N'utiliser que des *packages* installés par défaut (CTAN⁶⁵).
- Employer la feuille de style de l'éditeur si elle existe.
- Respecter au mieux la séparation du fond et de la forme.
 ➔ Éviter de jalonner le texte de commandes de mise en forme.

Dans le second cas, l'auteur doit seulement rendre un fichier PDF.

Il reste totalement maître du code source, de la façon de le gérer et de l'organiser, et concrètement de la nature des commandes et des *packages* utilisés.

2.2 Structure du document source

Structure L^AT_EX du document source

Tout fichier L^AT_EX a la structure suivante :

```
\documentclass[...]{}  Choix de la classe du document.
\usepackage[...]{}
...
\usepackage[...]{}      Appel des packages...

      AUTRES DÉCLARATIONS ... et des définitions générales globales.

\begin{document}
      CORPS DU TEXTE      Le corps du texte est contenu dans l'environnement
                        \begin{document} ... \end{document}.
\end{document}
```

Tout ce qui est au delà de `\end{document}` est ignoré par L^AT_EX.

On appelle préambule la zone comprise entre `\documentclass[-]{-}` et `\begin{document}`.

`\documentclass[-]{-}` : les classes

La classe d'un document détermine plusieurs points importants :

- son apparence ;
- sa structure logique.

Les classes standard de L^AT_EX :

article C'est la classe usuelle pour les articles scientifiques.

Des déclinaisons conçues pour des revues particulières existent : **amsart**⁶⁵ et **amsproc** (American Mathematical Society), **revtex**⁶⁶ (American Institute of Physics), **svjour**⁶⁷ (Springer Verlag Journals)...

Ces déclinaisons sont plus riches en métadonnées sur les auteurs et la nature du document. Les options de ces déclinaisons permettent de sélectionner le journal final, l'état *draft* ou *referee* (double interlignes)...

report C'est une classe prévue pour des documents plus conséquents avec chapitres et résumé. Peut convenir pour un rapport d'étudiant.

book Comme son nom l'indique, est spécialement prévue pour les livres (pas de résumé).

En fait, **report** peut aussi faire l'affaire pour des livres.

L'AMS fournit une déclinaison **amsbook** pour ses collections.

.....
`\documentclass[-]{-} : les classes (suite)`

Autres classes :

beamer C'est une classe fort riche qui permet des présentations par vidéo-projecteurs. C'est celle utilisée par ce document.

Nombreuses fonctionnalités : système de “pauses” pour faire apparaître le texte graduellement, environnements divers pour mise en valeur, gestion de thèmes (aspect général, éléments présents, couleurs...), préparation d'une version “article” de l'exposé...
 À consommer sans modération !

memoir Cette classe a été conçue par son auteur pour répondre à l'éternelle question de l'utilisateur usuel de \LaTeX : comment modifier l'aspect de tel ou tel élément du document ? Cette classe est basée à l'origine sur **book**, qu'elle étend de nombreuses fonctionnalités. C'est une solution “tout-en-un” à de nombreux problèmes, plutôt qu'une solution “modulaire” qui consisterait à ajouter une fonctionnalité grâce à un *package* approprié. De nombreux *packages* sont donc déjà “intégrés” dans **memoir**, sans risque d'incompatibilité. Le prix à payer : la taille du projet et de sa documentation !

.....
`\documentclass[-]{-} : les classes (suite)`

Autres classes :

Les classes de KOMA-script C'est un ensemble de classes et de *packages* associés qui remplacent les classes standard de \LaTeX : **scrbook**, **scrreprt**, **scrartcl**, **scr1ttr2**.
 L'idée est de fournir à l'utilisateur des commandes de personnalisation et un meilleur contrôle du style du document.

Autres Il existe d'autres classes plus confidentielles ou vouées à disparaître : **slides** remplacée par **beamer** ; **letter** qui n'est pas très efficace...

Dans ce cours, il sera essentiellement question des classes usuelles.

Les classes de KOMA-script et **memoir** seront ignorées.

.....
`\documentclass[-]{-} : les options de classes`

De nombreuses options modifient l'aspect du document :

Taille des caractères : 10pt, 11pt, 12pt.

Taille de la page : a4paper, a5paper, letterpaper...

Mise en page : twoside, landscape, onecolumn, twocolumn, openright, openany, titlepage.

Divers : draft, final, leqno, fleqn, openbib.

Certaines de ces options sont surclassées par des *packages* qui font mieux :

geometry permet de gérer la taille de la page et son orientation,

multicol gère les mises en page multi-colonnes,

amsmath propose des options fleqn, leqno et reqno,

titling personnalise la page de titre,

natbib redéfinit la gestion de la bibliographie,

draftwatermark s'occupe de signaler que le document est un “draft”...

Les classes proposées par les éditeurs de revues, **amsart**, **revtex** et **svjour**... proposent d'autres options personnalisées.

Certaines options proposées par les classes sont utilisées (et fournies) par des *packages* : **draft** et **final** sont proposées dans **graphicx** et **hyperref**, a4paper, a5paper, letterpaper... sont disponibles dans **geometry**...

.....
`\usepackage[-]{-} : les packages`

Les *packages* sont des modules qui ajoutent des fonctionnalités à \LaTeX .

➔ il faut donc savoir si un *package* fait ce qu'on désire, et lequel.

Un *package* contient un ensemble de macros dans un fichier .sty (ou plusieurs fichiers).

Une fois appelé par la commande `\usepackage{-}`, on a accès à des commandes et des environnements nouveaux.

On peut appeler plusieurs *packages* en même temps :

`\usepackage{pack1, pack2, ...}`

➔ regrouper les *packages* thématiquement proches.

Il y a des *packages* pour des usages très différents : encodage du texte source, gestion des polices, gestion de l'apparence des pages, inclusion de graphiques, fonctionnalités mathématiques avancées, gestion de langues, dessins internes au document, programmation...

Les *packages* peuvent comporter des options qu'on peut activer lors de l'appel.

La commande `\usepackage[english, french]{babel}` appelle le *package* **babel** avec les options `english` et `french`.

.....
`\usepackage[-]{-} : les packages (suite)`

⚠ Tous les *packages* sont égaux, mais certains sont plus égaux que d'autres.

- Certains *packages* modifient des fonctionnalités fournies par d'autres *packages*
 ➔ il est préférable de les appeler en dernier (**babel**, **hyperref**...).
- Des *packages* différents peuvent avoir des fonctionnalités semblables et entrer en conflit l'un avec l'autre.
- Un *package* peut en cacher un autre : il est possible qu'un *package* appelle d'autres *packages*, donc surveiller les incompatibilités induites.

Pour certains projets, on peut créer soi-même un *package* qui fera appel à d'autres *packages* et qui ajoutera certaines macros personnelles.

Des livres sur L^AT_EX permettent de découvrir l'existence de *packages* intéressants et d'en comprendre le fonctionnement.

Le site du CTAN⁷⁵ permet de chercher dans la description des *packages*.



Classes, styles, conflicts: The biological realm of L^AT_EX⁷⁶, par D. Verna.

..... Structure logique et sémantique d'un document

Chaque classe a sa propre structure logique.

La structure logique la plus utilisée est celle des découpages en parties, chapitres, sections, sous sections, paragraphes et sous paragraphes.

Les classes **report** et **book** définissent les commandes `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph`.

La classe **article** ne connaît pas `\chapter`.

Ces commandes acceptent un argument optionnel pour donner une version courte.

`\section[Titre court]{Version longue du titre qui n'en finit pas}`

Le titre long est celui utilisé dans le corps du texte.

Le titre court est utilisé à la fois dans la table des matières et dans les entêtes des pages.

Après le texte principal, on peut placer des annexes avec la commande `\appendix`.

Automatiquement la numérotation de plus haut niveau (`\section` pour **article**, `\chapter` pour **report** et **book**) devient alphabétique.

La classe **book** permet de diviser le texte en trois grandes parties :

une préface qui démarre avec `\frontmatter`, le corps du texte qui démarre avec `\mainmatter` et une post-face qui démarre avec `\backmatter`.

Les chapitres ne sont pas numérotés dans la préface et la post-face.

La numérotation des pages se fait en romain dans la préface.

..... Éléments gérés par L^AT_EX

L^AT_EX gère un certain nombre d'éléments automatiquement.

Numérotations : L^AT_EX s'occupe de numéroté de nombreux éléments : parties, chapitres,..., paragraphes, pages, notes de bas de page, tableaux, figures, équations, théorèmes (ou équivalents), items de listes...

Table des matières : En se basant sur les commandes `\chapter{-}`, `\section{-}`,... la commande `\tableofcontents` imprime une table des matières où elle est placée.

Selon la classe du document, cette table des matières démarre un nouveau chapitre (**report** et **book**) ou une section (**article**).

Il y a des commandes équivalentes pour la liste des tableaux (`\listoftables`) et la liste des figures (`\listoffigures`).

Références croisées : La commande `\label{lb}` permet de faire référence ailleurs dans le document au numéro actuel (qui dépend du contexte : chapitre, section, note de bas de page, équation, tableau, figure, théorème, item...) et à la page actuelle.

`\ref{lb}` imprime le numéro ainsi préservé,

`\pageref{lb}` imprime la page.

`lb` est interne au document et ne doit pas être utilisé deux fois dans `\label{-}`.

Bibliographie, Index : La gestion de la bibliographie peut être largement automatisée grâce à `bibtex`, et la gestion des index est réalisée grâce à `makeindex`.

Plus de détails seront donnés par la suite.

..... Les multiples compilations

Pour réduire l'utilisation de la mémoire vive, T_EX (et donc L^AT_EX) ne conserve en mémoire aucune donnée sur le texte lors d'une compilation.

➔ L^AT_EX utilise des fichiers auxiliaires dans lesquels il place des données à préserver : `.aux` par défaut, mais des *packages* en produisent d'autres.

Ces fichiers auxiliaires sont alors utilisés pour formater des fichiers "prêts à l'emploi" :

`.toc`, `.lof`, `.lot`, `.bbl`, `.ind`...

– Première étape : en compilant, L^AT_EX remplit de données les fichiers auxiliaires.

➔ `.aux` contient les données pour les références croisées, les entrées de la table des matières, les références bibliographiques utilisées...

– Certains moteurs doivent traiter certaines données : `bibtex` extrait les entrées bibliographiques d'une base de donnée `.bib` et produit un `.bbl`, `makeindex` formate les entrées de l'index sauvegardées dans `.idx` et produit un `.ind`...

➔ lancer ces moteurs si L^AT_EX a modifié les données de bibliographie ou d'index.

– Compilation suivante : L^AT_EX lit et met à jour les fichiers auxiliaires, utilise les fichiers "prêts à l'emploi" (`.toc`, `.bbl`, `.ind`...).

⚠ L^AT_EX s'occupe lui même de créer le fichier formaté `.toc` (table des matières) à partir du `.aux`

➔ il faut parfois 3 compilations pour stabiliser la table des matières !

Produire un gros document exige plusieurs compilations L^AT_EX, entrecoupées de l'usage de moteurs tels que `bibtex` et `makeindex`.

La console de compilation indique si le fichier `.aux` a été modifié.

Découper un document en pièces

Lorsqu'on gère un projet important, il est utile de le découper en pièces plus petites.

De façon générale, le document source peut demander à insérer d'autres documents.

`\input{fichier.tex}` lit le fichier `fichier.tex` à l'endroit désigné, l'interprète, et retourne ensuite à la suite du document courant.

- n'importe où dans le document (préambule ou corps du texte);
- on peut ainsi inclure une portion du texte final ou un ensemble de commandes;
- la recherche se fait dans les dossiers `texmf` comme décrit auparavant : le fichier n'est pas obligé d'être dans le dossier courant.

`\include{chap1}` dans le corps du texte inclut le fichier `chap1.tex` (\triangle extension).

Cette commande démarre une nouvelle page avant et après.

- réserver à des chapitres entiers de documents importants (livres, photocopiés...);
- un fichier `.aux` est créé pour chaque fichier inclus.

`\includeonly{chap1, chap3, chap4}` dans le préambule permet de sélectionner certains fichiers seulement.

2.3 Repères sur les concepts de programmation

Définir une commande de base

Avant même de découvrir des commandes, on va apprendre à en définir.

LaTeX fournit la commande `\newcommand{-}{-}` : le premier argument est le nom de la commande à définir; le second argument est le code définissant cette commande.

Le code peut être (presque) n'importe quoi.

- on peut y stocker du texte tout bête ou produire de grandes choses !

`\newcommand{\qed}{Quod erat demonstrandum}`

Cette commande produit "Quod erat demonstrandum" lorsqu'elle est invoquée.

`\newcommand{\formuleEinstein}{Formule d'Einstein : $E=mc^2$ }`

`\formuleEinstein` → Formule d'Einstein : $E = mc^2$

Pour définir une commande avec arguments, on utilise la syntaxe :

`\newcommand{-}{-}{-}` où `[-]` contient le nombre d'arguments.

Une commande ne peut pas accepter plus de 9 arguments.

`\newcommand{\xpy}[2]{ $\$1$ à la puissance $\$2$ s'écrit $\$1^{\$2}$ }`

`\xpy{3}{5}` → 3 à la puissance 5 s'écrit 3^5

`\xpy{(a+b)}{n}` → $(a + b)$ à la puissance n s'écrit $(a + b)^n$

Si une commande est déjà définie, on peut la redéfinir en utilisant `\renewcommand` à la place de `\newcommand`, avec la même syntaxe.

La création de commandes et d'environnements fera l'objet d'un autre cours (plus long).

Les dimensions

TeX peut manipuler des dimensions dans différentes unités :

mm	millimètre	cm	centimètre	in	inch	pt	point ⁶⁹
bp	big point	pc	pica ⁷⁰	dd	point Didot ⁷¹	cc	unité Cicéro ⁷²
sp	scaled point	ex	hauteur de "x"	em	largeur de "M"	mu	math unit

Quelques facteurs de conversions :

`1in = 2.54cm = 72.27pt = 72bp ; 1pc = 12pt ; 1157dd = 1238pt ; 1cc = 12dd`

Les unités `em` et `ex` dépendent de la fonte de caractères courante :

	1em	1ex	1pt	1cm	1in
<code>\tiny</code>	$\frac{1}{16}$	$\frac{1}{12}$	$\frac{1}{72}$	$\frac{1}{2.54}$	$\frac{1}{72.27}$
<code>\normalsize</code>	$\frac{1}{12}$	$\frac{1}{10}$	$\frac{1}{72}$	$\frac{1}{2.54}$	$\frac{1}{72.27}$
<code>\Large</code>	$\frac{1}{6}$	$\frac{1}{8}$	$\frac{1}{72}$	$\frac{1}{2.54}$	$\frac{1}{72.27}$

En plus des dimensions fixes, TeX manipule des dimensions "élastiques".

Leur syntaxe est de la forme `2cm plus 2mm minus 3mm`.

Ces "marges de manœuvre" sont utilisées pour ajuster les espaces horizontaux et verticaux.

Les dimensions sont des commandes qu'on ne peut pas invoquer seules :

`\parindent`, `\parsep`, `\baselineskip`, `\linewidth`, `\paperwidth`, `\textheight`,
`\tabcolsep`, `\topmargin`, `\itemsep`, `\labelsep`...

Jouer avec les longueurs

LaTeX fournit des commandes pour gérer les dimensions.

`\newlength{\len}` définit la nouvelle commande de dimension `\len`.

Des opérations simples sont possibles :

`\setlength{\len}{\langle dim \rangle}` assigne la valeur `\langle dim \rangle` à `\len`,

`\addtolength{\len}{\langle dim \rangle}` ajoute `\langle dim \rangle` à `\len`.

Il est possible d'afficher une dimension avec la commande `\the` :

```
\newlength{\len} \setlength{\len}{3cm} \the\len,
\addtolength{\len}{1mm plus 2pt minus 3pc} \the\len
85.35826pt, 88.20352pt plus 2.0pt minus 36.0pt
```

En interne, TeX travaille en points, et donc les dimensions sont affichées dans cette unité.

On peut récupérer la valeur des longueurs dans la console de compilation ou dans le fichier .log en utilisant la commande `\typeout{\len=\the\len}` qui produit `\len=85.35826pt`.

Il existe une dimension élastique particulière : `\fill` peut aller de 0pt à ∞.

`\the\fill` → 0.0pt plus 1.0fill

Cette “dimension” occupe l'espace disponible, comme un ressort en extension.

On peut augmenter la “dureté” du ressort.


La commande `\stretch{nbr}` vaut `nbr` fois `\fill` où `nbr` est un nombre entier :

`\the\stretch{4}` → 0.0ptplus4fill

Faire de l'espace horizontalement

TeX ignore les espaces entre objets. Il existe des commandes pour créer de vrais espaces.

`\hspace{\langle dim \rangle}` produit un espace horizontal de dimension `\langle dim \rangle` :

`a b\hspace{1cm}c` → `a b c` (1cm = )

En début de ligne, l'espace est ignoré.

On peut le forcer avec la version étoilée : `\hspace*{\langle dim \rangle}`

Des espacements horizontaux de taille prédéterminée sont fournis :

`\quad` est un espace de 1em : `a b`

`\enspace` est un espace de la moitié de `\quad` : `a b`

`\qqquad` est un espace du double de `\quad` : `a b`

`\hfill` est un espace de dimension `\fill` : `a b`

(l'espace entre `a` et `b` est le plus grand possible, `\hfill = \hspace{\fill}`)

`\hrulefill` comme `\fill` avec une ligne horizontale : `a b`

`\dotfill` comme `\fill` avec une ligne pointillée : `a b`

`\leftarrowfill` et `\rightarrowfill` : `a b c`

`\stretch{2}` est deux fois plus “puissant” que `\stretch{1}` (`=\fill`) :

`a\hspace{\stretch{1}}b\hspace{\stretch{1}}c` → `a b c`

`a\hspace{\stretch{1}}b\hspace{\stretch{2}}c` → `a b c`

On peut “doubler” la puissance d'un `\hrulefill` en en plaçant deux :

`a\hrulefill b\hrulefill\hrulefill c` → `a b c`

Faire de l'espace verticalement

`\vspace{\langle dim \rangle}` ajoute un espace vertical entre des paragraphes.

En début de page, cet espace vertical est supprimé :

il faut utiliser la version étoilée `\vspace*{\langle dim \rangle}`

⚠ il ne faut pas utiliser cette commande dans un paragraphe, mais entre des paragraphes.

LaTeX définit 3 commandes d'espacement verticaux :

`\smallskip` est un espace vertical de dimension `\smallskipamount`, qui vaut environ un quart de `\baselineskip`

`\the\smallskipamount` → 3.0pt plus 1.0pt minus 1.0pt

`\medskip` est un espace vertical de dimension `\medskipamount`, qui vaut environ la moitié de `\baselineskip`

`\the\medskipamount` → 6.0pt plus 2.0pt minus 2.0pt

`\bigskip` est un espace vertical de dimension `\bigskipamount`, qui vaut environ `\baselineskip`

`\the\bigskipamount` → 12.0pt plus 4.0pt minus 4.0pt

`\the\baselineskip` → 14.5pt

⚠ la longueur `\baselineskip` n'est pas absolue, elle dépend de certains choix effectués dans le document, en particulier de la hauteur des caractères.

Les compteurs

Le langage de TeX contient la notion de compteurs.

Les compteurs ne sont pas des commandes. Ce sont des variables utilisées à de nombreux endroits : numérotation des pages, des formules, des items de listes...

Les compteurs par défaut définis par LaTeX sont les suivants :

part	paragraph	figure	enumi
chapter	subparagraph	table	enumii
section	page	footnote	enumiii
subsection	equation	mpfootnote	enumiv
subsubsection	tocdepth		

`mpfootnote` est le compteur des *footnotes* dans l'environnement `minipage`, les compteurs `enumi` interviennent dans les listes.

`\newtheorem` définit des compteurs pour les théorèmes et autres structures sémantiques des mathématiques, d'autres *packages* peuvent fournir des compteurs...

À chaque compteur est associé une commande du type `\thectr` qui affiche le compteur, où `ctr` est le nom du compteur : `\thepage` → 37

Cette commande peut afficher plus que la valeur du compteur, par exemple `\theequation` peut rappeler le numéro de la section courante en affichant 2.4, où 2 est la valeur du compteur de section et 4 la valeur du compteur des équations.

Jouer avec les compteurs

L^AT_EX fournit des commandes pour gérer les compteurs.

`\newcounter{ctr} [masterctr]` définit le nouveau compteur *ctr*.

L'argument optionnel *masterctr* est un compteur “maître” existant : *ctr* est remis à 0 lorsque *masterctr* change.

Des opérations simples sont possibles :

`\setcounter{ctr}{val}` assigne la valeur *val* à *ctr*,

`\addtocounter{ctr}{val}` incrémente *ctr* de la valeur *val*,

`\stepcounter{ctr}` incrémente *ctr* d'une unité.

Le contenu d'un compteur est directement affichable sous différentes formes :

```
\arabic{-} \roman{-} \Roman{-} \alph{-} \Alph{-} \fnsymbol{-}
```

`\fnsymbol{-}` utilise une suite de 9 symboles * † ‡ § ¶ || ** †† ‡‡ en général utilisés pour la numérotation des notes de bas de page.

```
\newcounter{ctr} \setcounter{ctr}{4} \arabic{ctr}, \roman{ctr},
\Roman{ctr}, \alph{ctr}, \Alph{ctr}, \fnsymbol{ctr}.
```

4, iv, IV, d, D, §.

La commande `\value{ctr}` retourne la valeur du compteur *ctr* en tant que nombre

➔ utile dans les commandes qui demandent des nombres, par exemple comme second argument de `\setcounter`.

Redéfinir l'affichage d'un compteur

Les commandes `\thectr` peuvent être redéfinies par l'utilisateur.

```
\renewcommand{\theequation}{\arabic{section}-\Alph{equation}}
```

affiche les numéros d'équations sous la forme 2-D, où le compteur *section* vaut 2 et *equation* vaut 4.

```
\renewcommand{\theequation}{\thesection-\Alph{equation}}
```

réutilise l'affichage par défaut du compteur des sections.

➔ meilleure solution car respecte l'aspect d'un compteur déjà défini.

```
\renewcommand{\thesubsection}%
```

```
{\bfseries\thesection\alph{subsection}}
```

affiche les numéros des sous sections sous la forme 2c.

(Noter l'usage d'un caractère % pour découper cette redéfinition en plusieurs lignes sans ajouter d'espace supplémentaire.)

Le package **amsmath** définit la commande `\numberwithin{-}{-}` qui assujettit le compteur donné dans le premier argument au compteur donné dans le second :

`\numberwithin{subsection}{section}` remet le compteur des sous sections à 0 à chaque début de section et l'affichage est du type 2.3.

Les compteurs doivent exister avant d'invoquer cette commande.

2.4 Tour d'horizon de commandes de base

La justification

Par défaut, L^AT_EX justifie le texte dans les paragraphes : en jouant sur les espacements entre mots et les césures de mots, le texte est aligné sur les bords gauches et droits.

Les environnements `flushleft`, `flushright` et `center` désactivent les espacements “élastiques” entre les mots et les césures pour, respectivement, coller le texte sur le bord gauche, coller le texte sur le bord droit et centrer le texte.

Ces environnements acceptent des paragraphes et des ruptures de lignes avec `\\`.

```
\begin{center}
Actibus immensis\\
urbs fulget\\[2mm]
massiliensis
\end{center}
```

Ces environnements ont des commandes (presque) équivalentes : `\raggedright`, `\raggedleft` et `\centering` (⚠ noms inversés !).

```
{\raggedleft Felix, qui potuit
rerum cognoscere causas\par}
Felix, qui potuit rerum
cognoscere causas
```

⚠ ces commandes ne fonctionnent que sur un paragraphe entier, d'où le `\par`.

Les listes

L^AT_EX définit trois types de listes : listes à puces, listes numérotées et listes à mots clés.

```
\begin{itemize}
\item In vino veritas – In vino veritas
\item Natura abhorret a vacuo – Natura abhorret a vacuo
\end{itemize}
```

```
\begin{enumerate}
\item Qui rogat, non errat 1. Qui rogat, non errat
\item Qui scribit, bis legit 2. Qui scribit, bis legit
\end{enumerate}
```

```
\begin{description}
\item[Cicéron :] Cuiusvis hominis est errare
\item[Horace :] Qui cupit aut metuit liber non erit unquam
\end{description}
```

L'environnement de liste “brut”, `list`, permet de définir d'autres types de listes.

Des environnements de type listes se cachent ailleurs : les environnements de bibliographie, d'index, de théorèmes...

Les listes (suite)

On peut emboîter jusqu’à 4 niveaux de listes :

```
\begin{enumerate}
  \item UN
    \begin{enumerate}
      \item un de UN
        \begin{enumerate}
          \item premier
          \item second
        \end{enumerate}
      \item deux de UN
    \end{enumerate}
  \item DEUX
\end{enumerate}
```

Les compteurs `enumi`, ..., `enumiv` correspondent à ces emboîtements.

Les commandes `\labelenumi...` et `\labelitemi...` interviennent dans l’affichage des marques d’items, respectivement numérotés et non numérotés.

`\renewcommand{\labelitemii}{+}` place un “+” comme marque d’item au niveau 3 de `itemize`, `\renewcommand{\labelenumii}{\theenumi-\theenumii}` redéfinit l’aspect des numéros au niveau 2 de `enumerate`.

Les tabulations

L’environnement `tabbing` définit un système de tabulations.

```
\begin{tabbing}
AAAAAA \= BBBB \= CCCC \\\
DD \> EE \> FF \\\
G \> H \> I \\\[3mm]
AAA \= BB \> CC \\\
à \> é \> ê
\end{tabbing}
```

`\=` place un taquet, `\>` positionne le texte qui suit sur le taquet suivant.

```
\begin{tabbing}
\hspace*{2cm}\= \hspace{1cm}\= \kill
a \> b \> c
\end{tabbing}
```

`\kill` n’affiche pas la ligne (→ positionnement de taquets).

D’autres commandes de tabulations sont disponibles, `\’`, `\’`, `\+`, `\-`, dont le comportement est trop long à expliquer ici.

⚠ Ne pas utiliser `\’` et `\’` pour les lettres accentuées dans cet environnement.

Le package **Tabbing** règle ce problème.

Les tableaux

L’environnement `tabular` permet de composer des tableaux.

```
a \begin{tabular}[t]{|rc|l|}
ABC & DEF & GHI \\ \hline
J & K & L \\ \hline
\multicolumn{2}{|c|}{MNO} & PQRS \\ \end{tabular} b
```

Les colonnes peuvent être du type `r`, `c`, `l`, `p{<dim>}` (paragraphe de largeur `<dim>`).

L’argument optionnel de `tabular` gère l’alignement vertical du tableau.

On peut insérer du texte entre deux colonnes avec la syntaxe `r@{texte}l`

```
a \begin{tabular}[c]{r@{=}l}
d & 2 \\
c & 3 \\ \end{tabular} b
```

Différentes dimensions gèrent l’aspect des tableaux : `\tabcolsep` (séparation entre colonnes), `\arrayrulewidth` (épaisseur des traits), `\doublerulesep` (séparation entre les lignes doubles), `\arraystretch` est un facteur multiplicatif de l’espace entre les colonnes.

La gestion avancée des tableaux fera l’objet d’un autre cours.

Les flottants

Les éléments flottants sont des objets dont le placement exact dans le document est laissé à la sagacité de `LaTeX`. Les flottants sont conservés en mémoire qu’à rencontrer un endroit adéquat pour les placer (dans l’ordre d’arrivée).

→ parfois cet endroit ne se rencontre qu’à la fin du document.

→ un flottant peut bloquer le placement des flottants suivants.

Par défaut, `LaTeX` définit deux environnements flottants : `figure` et `table`.

`figure` s’utilise pour les graphiques, et `table` s’utilise pour les tableaux.

⚠ Pour insérer un graphique ou un tableau, on n’a pas besoin de ces environnements !

```
\begin{figure}[h]
matériel de la figure
\caption[Légende courte]{Un longue légende qui n'en finit pas}
\label{fig-label}
\end{figure}
```

Ces deux environnements ont pour fonctions :

1. de placer leur contenu à un endroit du document où il y a suffisamment de place ;
2. de placer une légende numérotée renseignée par la commande `\caption[-]{-}`, de type `Figure` ou `Tableau` ;
3. de conserver les données (légende, page, numéro) dans un fichier auxiliaire pour construire une table des figures ou une table des tableaux.

.....

Les flottants (suite)

L'option de ces environnements impose un certain comportement pour placer le flottant dans la page : t (haut de page), b (bas de page), p (page de flottants), h (ici si possible).

Ces options peuvent être combinées dans l'ordre de préférence : [htp].

L'option p place le flottant sur une page *réservée* aux flottants.

Le package **Float** ajoute l'option H qui place le flottant exactement à l'endroit défini.

Différents paramètres globaux (compliqués) permettent de gérer la façon dont les flottants sont placés dans le document.

`\topfraction` : proportion max. de la page réservée aux flottants en haut.

→ `\renewcommand{\topfraction}{0.9}` permet de placer des flottants plus nombreux ou plus grands (0.7 par défaut).

`\bottomfraction` : proportion max. de la page pouvant contenir des flottants en bas.

`\textfraction` : proportion min. de la page contenant du texte.

→ `\renewcommand{\textfraction}{0.1}` permet d'accepter plus de flottants (0.2 par défaut).

Voir aussi `\topnumber`, `\bottomnumber`, `\totalnumber`, `\floatpagefraction`.

La commande `\clearpage` (ou `\cleardoublepage` en mode twoside), interrompt la page en court, *place tous les flottants en attente*, et crée une nouvelle page.

→ cette commande est appelée pour créer un nouveau chapitre par exemple.

La gestion avancée des flottants fera l'objet d'un autre cours.

.....

La page de garde

Dans certaines classes, L^AT_EX compose une page de garde à partir de données spécifiques :

`\title{-}` définit le titre, `\author{-}` renseigne les auteurs (séparés par la commande `\and`), `\date{-}` insère la date. Des commandes `\thanks{-}` associées aux auteurs produisent des informations en bas de page.

Après ces commandes, `\maketitle` compose le titre avec les éléments fournis.

Dans certaines classes, l'environnement `abstract` met en forme un résumé.

Les classes **amsart** et **amsproc** proposent des modifications : `\title[-]{-}` (titre court en option), `\translator{-}`, `\dedicatory{-}`, `\author[-]{-}` (nom court en option), `\contributor[-]{-}`, `\address{-}` (adresse postale), `\curraddr{-}` (adresse courante), `\email{-}`, `\urladdr{-}` (site WEB), `\thanks{-}`, `\subjclass[-]{-}` (**Mathematics Subject Classification** ⁷³), `\keywords{-}`...

La classe **revtex** propose les commandes suivantes : `\surname{-}` (pour l'indexation alphabétique), `\email[-]{-}`, `\homepage[-]{-}` (page WEB), `\affiliation{-}` (adresse), `\altaffiliation[-]{-}` (adresse temporaire), `\collaboration{-}`, `\pacs{-}` (**Physics and Astronomy Classification Scheme** ⁷⁴), `\keywords{-}`, `\preprint{-}`...

La classe **svjour** propose les commandes suivantes : `\subtitle{-}`, `\dedication{-}`, `\institute{-}`, `\email{-}`...

→ on est loin d'une normalisation des métadonnées !

.....

La bibliographie

L'environnement `thebibliography` démarre un nouveau chapitre ou une nouvelle section (selon la classe) et installe une liste pour présenter des références bibliographiques :

```
\begin{thebibliography}{00}
  \bibitem{ChamConn07} Ali~H. Chamseddine and Alain Connes.
  \newblock {Why the Standard Model}.
  \newblock {\em Journal of Geometry and Physics}, 58:38–47, 2008.
```

```
\bibitem{Wei67}{Wein67a} Stephen Weinberg.
  \newblock A model of leptons.
  \newblock {\em Phys. Rev. Lett.}, 19(21):1264–1266, Nov 1967.
\end{thebibliography}
```

Les entrées sont libellées en interne par l'argument obligatoire de `\bibitem[-]{-}`.

Par défaut, les entrées ont un libellé apparent sous forme d'un numéro (dans l'ordre).

L'argument optionnel produit un libellé apparent personnalisé.

L'argument obligatoire de l'environnement `thebibliography` sert à fixer la largeur maximale des libellés apparents (seul le nombre de lettres compte !).

On peut citer une référence bibliographie avec `\cite[-]{-}` :

```
\cite{ChamConn07} → [1], \cite{Wein67a,ChamConn07} → [Wein67, 1],
\cite{Thm~4}{ChamConn07} → [1, Thm 4]
```

La gestion avancée de la bibliographie (avec `bibtex`) fera l'objet d'un autre cours.

.....



Commandes en vrac : gestion de la mise en page

Quelques commandes de mise en page :

- `\newpage` impose un saut de page.
- `\pagebreak[num]` suggère un saut de page.
De $num = 0$ (on peut...) à $num = 4$ (on veut...).
- `\nopagebreak[num]` dissuade un saut de page.
- `\clearpage` traite les flottants en court et impose un saut de page.
- `\cleardoublepage` procède de même en commençant une nouvelle page à droite (mode twoside).
- `\enlargethispage{<dim>}` agrandit la page courante de la dimension spécifiée.
- `\indent` crée l'indentation en début de paragraphe (largeur `\parindent`).
- `\noindent` supprime l'indentation en début de paragraphe.
La dimension `\parindent` correspond à cette indentation :
→ `\setlength{\parindent}{0pt}` supprime toute indentation.
- `\newline` interrompt la ligne courante et en démarre une nouvelle.
- `\linebreak[num]` suggère une interruption de ligne.
- `\nolinebreak[num]` dissuade une interruption de ligne.
- `\footnote{-}` insère une note de bas de page.

.....
Commandes en vrac : commandes de typographie

Quelques commandes de nature typographique :

- Les environnements `quote` et `quotation` servent à citer du texte : retraits du texte à droite et à gauche, `quotation` indente les paragraphes.
- L’environnement `verse` est destiné à la poésie.
- L’environnement `verbatim` sert à présenter du texte non interprété.
- Les guillemets (anglais) sont saisis avec la syntaxe ‘ ‘mot’ ’ \Rightarrow “mot”
- Il y a trois sortes de tirets :
 - \Rightarrow - (trait d’union), -- \Rightarrow - (tirets de listes), --- \Rightarrow — (début de dialogue)
- Les points de suspension sont obtenus pas `\dots` ou `\ldots` \Rightarrow ...
- Césures : \TeX dispose d’une liste de césures.
On peut localement l’aider en découpant un mot soi-même : `al-gè-bre`
On peut le faire globalement : `\hyphenation{al-gè-bre vec-to-riel}`
`\showhyphens{-}` affiche dans la console et le `.log` les suggestions de \TeX :
`\showhyphens{algèbre vectoriel}` \Rightarrow al-gèbre vec-to-riel
- “Pavé noir” : `\rule[$\langle dimh \rangle$]{ $\langle dimx \rangle$ }{ $\langle dimy \rangle$ }` créé un bloc plein de taille $\langle dimx \rangle \times \langle dimy \rangle$ relevé de la hauteur $\langle dimh \rangle$:
`\rule{1cm}{3pt}` \Rightarrow ,
`x\rule[1ex]{1cm}{1pt}x` \Rightarrow 

Cours 3 – Des packages incontournables

“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.

3.1 Le package inputenc	47
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...	
3.2 Le package fontenc	49
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...	
3.3 Le package textcomp	52
Où l’on découvre que l’alphabet n’est pas tout et qu’il est aussi possible de communiquer avec des signes bizarres...	
3.4 Le package lmodern	53
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...	
3.5 Le package geometry	54
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...	
3.6 Le package babel	56
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...	
3.7 Le package hyperref	57
Où l’on découvre qu’un document écrit peut se révéler dynamique sous les cliquetis d’une simple souris...	
3.8 Les packages amssymb et amsmath	59
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...	
3.9 Le package graphicx	60
Où l’on apprend qu’on peut faire usage du vieil adage qui prétend qu’un bon dessin vaut mieux qu’un long discours...	

.....
Une modèle de fichier source

Voici à quoi peut ressembler un fichier source L^AT_EX de base :

```
\documentclass[12pt]{article}
\usepackage[latin1]{inputenc}
\usepackage[T1,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**...

3.1 Le *package* **inputenc**

.....
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 :
 $\backslash\alpha \rightarrow \alpha$, $\backslash\text{copyright} \rightarrow \copyright$, et les lettres accentuées peu usuelles...
 L^AT_EX 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'ASCII⁷⁵, 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 \backslash' e pour encoder é.

Le choix de compléter ce tableau à 256 caractères dépend des systèmes d'exploitations :
 Windows (ansi⁷⁶), Macintosh (apple⁷⁷) et Unix/Linux (latin1⁷⁸) 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

UNICODE⁷⁹ 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*⁸⁰, l'écriture inventée par J. R. R. Tolkien et qui figure en bonne place dans **Le Seigneur des Anneaux**⁸¹.

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

(Voir la page Wikipédia⁵⁵ et les **Tableaux de caractères Unicode**⁸².)

Interlude : UNICODE (suite)

Extrait de Introduction à Unicode et à l'ISO 10646 par Patrick ANDRIES :

Table with 3 columns: Glyphes représentatifs, Numéro, Nom officiel français de l'ISO/CEI 10646. Rows include Latin letters (D, v, o, f, a, k), Space, Chinese (天, 心), Arabic (س, ج, ا, م), Greek (α), Gothic (ψ), Musical (c), and Tibetan (A, -).

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 ! É l'Ždition, le fichier semble rempli de hiŽroglyphes bizarres ! (encodage apple mac 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.

3.2 Le package fontenc

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 morcelée 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[T1,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`.

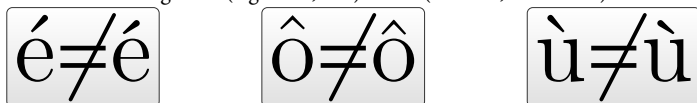
Pour résoudre ces problèmes, il suffit de passer en encodage T1 (pour les langues latines).

Les polices d'écriture par défaut de \TeX sont les déclinaisons de COMPUTER MODERN (CM). Elles ont été créées par D. Knuth en même temps que \TeX . 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) :



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 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 ôta son chapeau
devant le médecin hébété.
Il ôta son chapeau devant le
médecin 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[T1,T1]{fontenc}
```

ou

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

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

3.3 Le package `textcomp`

.....
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 :

£ ¤ § ¶ ¸ © ® % ¯ º ± • * ™ ☞

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-`.

`$45\tcdegree$`, `$100\, \tc{ohm}$` 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>	∕o∕

3.4 Le package `lmodern`

.....
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 accentuées 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** :

Υ Œ Ê Æ Ñ Ñ Ñ Ñ

(Extrait de **An exploration of the Latin Modern fonts** ⁸⁴ 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 L^AT_EX à utiliser les polices **CM-SUPER** qui n'ont pas de grandes qualités typographiques.

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

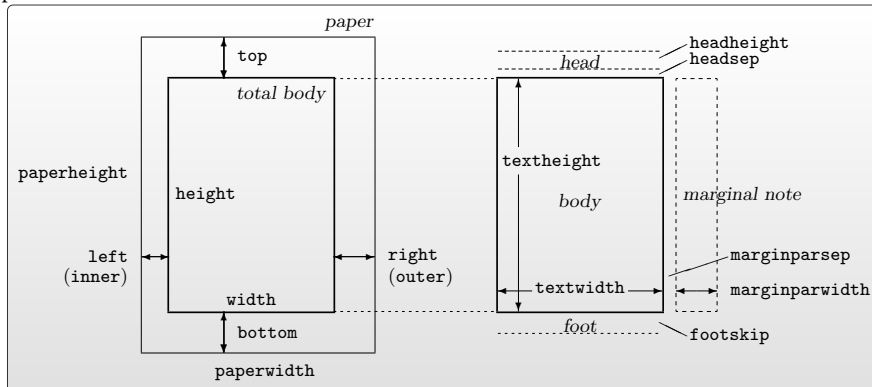
Enfin, les polices **LATIN MODERN** sont disponibles en **OPENTYPE**, ce qui les rend accessibles à n'importe quel logiciel “ordinaire” sur l'ordinateur.

➡ Pratique pour externaliser la production d'un morceau de document tout en conservant la typographie de T_EX.

3.5 Le *package geometry*

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=*<dim>*, paperheight=*<dim>*...

Taille du corps textwidth=*<dim>*, textheight=*<dim>*, text=*<dim>*, *<dim>*...

Taille des marges tmargin=*<dim>*, ..., hmargin=*<dim>*, vmargin=*<dim>*, inner=*<dim>*, outer=*<dim>*...

➔ 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.

3.6 Le *package* `babel`

Au tour de `babel`

LaTeX 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 LaTeX 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` ➔ N^{os}...

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.

3.7 Le *package* `hyperref`

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 LaTeX (`\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 \LaTeX , 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*.

3.8 Les *packages* `amssymb` et `amsmath`

Mathématiques supérieures avec `amssymb` et `amsmath`

Par défaut, \LaTeX s’occupe très bien des mathématiques.

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

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

`\varnothing` → \emptyset , `\bigstar` → \star , `\boxplus` → \boxplus , `\ltimes` → \ltimes .

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

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

`\mathfrak{-}` 012...789 $\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{D}\dots\mathfrak{X}\mathfrak{Y}\mathfrak{Z}$ $\mathfrak{a}\mathfrak{b}\mathfrak{c}\mathfrak{d}\dots\mathfrak{x}\mathfrak{y}\mathfrak{z}$
`\mathbb{-}` $\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\dots\mathbb{U}\mathbb{V}\mathbb{W}\mathbb{X}\mathbb{Y}\mathbb{Z}$ (⚠ 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 `\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.

3.9 Le package `graphicx`

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.

Cours 4 – La police a bon caractère

"I can't go to a restaurant and order food because
I keep looking at the fonts on the menu."
Donald Knuth
"All Questions Answered". *Notices of the AMS* 49 (3): 321.

4.1 Les commandes de base	62
Où l'on prend du galon pour commander à la police...	
4.2 Commandes plus avancées	66
Où l'on apprend quelques subtilités sur l'art des fontes et où l'on pénètre dans les secrets d'une police...	
4.3 Autres polices d'écriture et symboles divers	71
Où l'on s'étonne de pouvoir se faire obéir par des glyphs très divers dont on ne comprend pas toujours la signification...	
4.4 Les commandes de police en mathématique	74
Où l'on découvre diverses façons d'insérer du texte au sein de formules, et où l'on s'éduque à l'art difficile de la graisse épaisse en mode mathématique...	

4.1 Les commandes de base

Les trois familles de polices de L^AT_EX

L^AT_EX est beaucoup plus riche en commandes que T_EX pour gérer le style des caractères.
Une police de caractères est d'abord une famille (*family*) de différentes fontes (variantes).
En interne, L^AT_EX installe 3 familles de polices :
Famille romane : C'est la police d'écriture utilisée par défaut dans le document.
Famille sans empattements : Cette police sert occasionnellement, par exemple pour des titres, des petites portions de texte à isoler du reste...
Famille à écartement fixe : C'est une police où l'espacement entre les lettres est constant, contrairement aux deux précédentes, où l'espacement est dit proportionnel.
On l'appelle aussi *typewriter* car l'espacement est fixe sur les machines à écrire.

Ces trois familles sont coordonnées pour donner un aspect agréable au document.
Exemples de combinaisons de trois familles :

Latin Modern Roman / Latin Modern Sans / Latin Modern Typewriter

Lucida Bright / Lucida Casual / Lucida Typewriter

Libertine / Biolinum / CM-Super Typewriter

Times / Helvetica / Courier

⚠ La répartition *roman/sans serif/typewriter* n'est pas nécessairement respectée, au sens où rien n'oblige à utiliser une police romane pour la première, une police sans empattements pour la seconde et une police à largeur fixe pour la troisième !

Encodage, graisses, formes et tailles

Chaque famille est déclinée en plusieurs graisses (*series*) :
principale (*medium*), épaisse (*bold*), semi-épaisse (*semi bold*, *bold condensed*...), légère (*light*)...
Chaque graisse peut se subdiviser en plusieurs formes (*shapes*) :
droite (*up*), italique (*italic*), penchée (*slanted*, *oblique*), petites capitales (*small caps*)...
Chacune de ces combinaisons peut alors se décliner en différentes tailles (*size*) :
très petit à **très grand** en passant par petit et grand...

Désigner une fonte de caractère revient donc à définir une combinaison
`encoding-family-series-shapes-size`

où l'encodage (*encoding*) est celui désigné par **fontenc**.
Exemples avec les polices LATIN MODERN (encodage T1, taille normale) :
Roman / *Roman Italic* / *Roman Oblique* / ROMAN SMALL CAPS
Roman Bold Extended / ***Roman Bold Oblique Extended***
Roman Bold Condensed / ***Roman Bold Oblique Condensed***
Sans Demi Condensed / *Sans Demi Condensed Oblique*
Typewriter Light / *Typewriter Light Oblique*

⚠ Certaines combinaisons peuvent ne pas exister en fonction de la police utilisée.

Les commandes de changement de style du texte

Commande	Attribut	Exemple
<code>\rmfamily</code>	<code>\textrm{-}</code>	family Roman
<code>\sffamily</code>	<code>\textsf{-}</code>	family Sans serif
<code>\ttfamily</code>	<code>\texttt{-}</code>	family Type Writer
<code>\mdseries</code>	<code>\textmd{-}</code>	series medium
<code>\bfseries</code>	<code>\textbf{-}</code>	series bold
<code>\upshape</code>	<code>\textup{-}</code>	shape up
<code>\itshape</code>	<code>\textit{-}</code>	shape <i>italic</i>
<code>\slshape</code>	<code>\textsl{-}</code>	shape <i>slanted</i>
<code>\scshape</code>	<code>\textsc{-}</code>	shape SMALL CAPS
<code>\normalfont</code>	<code>\textnormal{-}</code>	medium, up

Des *packages* peuvent définir d'autres commandes que celles-ci en fonction de ce que contient la police courante.
Les commandes `\normalfont` et `\textnormal{-}` rechargent les valeurs par défaut de l'encodage, de la famille, de la graisse et de la forme (voir plus loin).

Les commandes `\emph{-}` et `\em`

La commande `\emph{-}` met en valeur le texte qu'elle contient en basculant entre les formes droites et les formes italiques.
Elle admet un équivalent sous la forme `{\em ...}`.

`\sffamily\mdseries\upshape Veritatis \emph{simplex} oratio est`
Veritatis simplex oratio est

`\sffamily\bfseries\upshape Veritatis \emph{simplex} oratio est`
Veritatis simplex oratio est

`\itshape Veritatis \emph{simplex} oratio est`
Veritatis simplex oratio est

`\bfseries\itshape Veritatis \emph{simplex} oratio est`
Veritatis simplex oratio est

Les commandes de changement de la taille du texte

Commande	Exemple
<code>\tiny</code>	<small>tiny</small>
<code>\scriptsize</code>	<small>scriptsize</small>
<code>\footnotesize</code>	<small>footnotesize</small>
<code>\small</code>	<small>small</small>
<code>\normalsize</code>	<small>normalsize</small>
<code>\large</code>	large
<code>\Large</code>	Large
<code>\LARGE</code>	LARGE
<code>\huge</code>	huge
<code>\Huge</code>	Huge

Ce sont les commandes prédéfinies, mais il existe une façon d'accéder directement à n'importe quelle taille de police.

Exemples de changement de style et de taille

```
\sffamily\bfseries\slshape Veritatis simplex oratio est
Veritatis simplex oratio est
```

```
\rmfamily\bfseries\itshape Veritatis simplex oratio est
Veritatis simplex oratio est
```

```
\sffamily\bfseries\itshape Veritatis simplex
\normalfont oratio est
Veritatis simplex oratio est
```

```
\sffamily\large Veritatis
{\rmfamily\itshape{\bfseries\footnotesize simplex} oratio} est
Veritatis simplex oratio est
```

Noter sur cet exemple la portée des commandes.

Exemples de changement de style et de taille (suite)

```
\sffamily\large\bfseries\itshape Veritatis simplex oratio est
=
\itshape\large\bfseries\sffamily Veritatis simplex oratio est
Veritatis simplex oratio est = Veritatis simplex oratio est
```

L'ordre des commandes ne change pas le résultat.

⚠ Il faut bannir les anciennes commandes L^AT_EX2.09 : `\rm`, `\sf`, `\tt`, `\bf`, `\sl`, `\it`, `\sc`.

```
{\it\bf test}$\neq$\bf\it test}$\neq${\itshape\bfseries test}
test≠test≠test
```

Le résultat est incorrect et dépend de l'ordre des commandes !

4.2 Commandes plus avancées

Nommer les familles, les graisses et les formes

LaTeX désigne une famille par un nom court assez peu évocateur.
Exemples :

LATIN MODERN ROMAN	lmr	LATIN MODERN SANS	lms
LATIN MODERN TYPEWRITER	lmtt	LIBERTINE	fxl
BIOLINUM	fxb	GARAMOND	padx
TIMES	ptm	PALATINO	ttl
AVANT GARDE	pag	HELVETICA	phv
URW ANTIQUA	uaq	URW GROTESK	ugq
VENTURIS	yvt	CHARTERBT	bch

C'est ce nom qui est utilisé pour basculer d'une famille à une autre.
La codification des noms des polices est décrite sur le site [Filenames for TeX fonts](#)⁸⁵.

Les graisses et les formes sont aussi codifiées :

Graisses		Formes
<i>ultra light</i> ul	<i>light</i> l	<i>upright</i> n
<i>medium</i> m	<i>semibold</i> sb	<i>italic</i> it
<i>bold</i> b	<i>bold extended</i> bx	<i>slanted</i> sl
<i>condensed</i> c	...	<i>small caps</i> sc

Commandes de basculement

On peut basculer l'encodage, la famille, la graisse et la forme avec les commandes `\fontencoding{-}`, `\fontfamily{-}`, `\fontseries{-}`, `\fontshape{-}` qu'il faut ensuite activer avec `\selectfont`.
On peut aussi utiliser le raccourci `\usefont{-}{-}{-}{-}` avec les 4 attributs dans l'ordre (le `\selectfont` est inutile).

```
{\fontencoding{OT1}\fontfamily{ptm}\fontseries{b}\selectfont texte}
```

sélectionne la graisse épaisse de la fonte TIMES dans l'encodage OT1.

```
{\usefont{T1}{lmtt}{l}{sl} texte}
```

sélectionne LATIN MODERN TYPEWRITER LIGHT OBLIQUE.

```
{\fontseries{b}\fontshape{sl}\selectfont texte}
```

sélectionne la graisse épaisse dans la forme oblique de la famille courante.

⚠ Ne pas oublier de terminer la liste avec `\selectfont`.
Le changement ne prend effet que dans le groupe balisé par les deux accolades.

Les valeurs par défaut

LaTeX conserve les réglages par défaut des polices dans différentes commandes :

<code>\encodingdefault</code>	encodage par défaut	<code>\familydefault</code>	famille par défaut
<code>\seriesdefault</code>	graisse par défaut	<code>\shapedefault</code>	forme par défaut

Ces valeurs sont activées par les commandes `\normalfont` et `\textnormal{-}`.
`\rmdefault`, `\sfdefault` et `\ttdefault` contiennent les 3 familles par défaut.
`\mddefault` et `\bfdefault` contiennent les graisses par défaut.
`\updefault`, `\itdefault`, `\sldefault` et `\scdefault` les formes par défaut.
Au départ, `\familydefault` vaut `\rmdefault`.
`\sffamily` est équivalent à `\fontfamily{\sfdefault}\selectfont` ;
`\bfseries` est équivalent à `\fontseries{\bfdefault}\selectfont` ;
`\slshape` est équivalent à `\fontshape{\sldefault}\selectfont` ;
etc

Valeurs pour ce document : `\mddefault = m`, `\bfdefault = bx`, `\updefault = n`, `\itdefault = it`, `\sldefault = sl` et `\scdefault = sc`.
Pour changer ces valeurs, il faut utiliser des commandes du type `\renewcommand{\bfdefault}{b}`
On peut opérer de même avec toutes les commandes "default".

Commandes de changement de la taille

Le premier argument de la commande `\fontsize{-}{-}` fixe la taille de la police en points, et le second fixe la hauteur minimale (en points) de l'interligne (`\baselineskip`).

```
\fontsize{16}{18}\selectfont Du texte en 16pt  
Du texte en 16pt
```

```
\fontsize{8}{8}\selectfont : Dum apud Persas, ut supra narravimus, perfidia regis motus agit at insperatos, et in  
eois tractibus bella rediviva consurgunt, anno sexto decimo et eo diutius post Nepotiani exitium, saeviens per urbem aeternam urebat  
cuncta Bellona.
```

```
\fontsize{8}{12}\selectfont : Dum apud Persas, ut supra narravimus, perfidia regis motus agit at insperatos, et in  
eois tractibus bella rediviva consurgunt, anno sexto decimo et eo diutius post Nepotiani exitium, saeviens per urbem aeternam urebat  
cuncta Bellona.
```

L'argument de la commande `\linespread{-}` multiplie la valeur de `\baselineskip` :

```
\linespread{1.1}\selectfont :  
Dum apud Persas, ut supra narravimus, perfidia regis motus agit at insperatos, et  
in eois tractibus bella rediviva consurgunt, anno sexto decimo et eo diutius post  
Nepotiani exitium, saeviens per urbem aeternam urebat cuncta Bellona.
```

```
\linespread{1.5}\selectfont :  
Dum apud Persas, ut supra narravimus, perfidia regis motus agit at insperatos, et  
in eois tractibus bella rediviva consurgunt, anno sexto decimo et eo diutius post  
Nepotiani exitium, saeviens per urbem aeternam urebat cuncta Bellona.
```

.....

La face cachée des polices LATIN MODERN

La face cachée des polices LATIN MODERN (suite)

.....

Les polices LATIN MODERN se déclinent en différentes familles, graisses et formes.

La famille Roman de Latin Modern

```
\rmfamily
\mdseries
\upshape      LM Roman
\itshape      LM Roman Italic
\slshape      LM Roman Oblique
\scshape      LM ROMAN SMALL CAPS
\fontshape{scs1} LM ROMAN SMALL CAPS OBLIQUE
\fontshape{ui}  LM Roman Unslanted italic
\bfseries
\upshape      LM Roman Bold Extended
\itshape      LM Roman Bold Italic Extended
\slshape      LM Roman Bold Oblique Extended
\fontseries{b}
\upshape      LM Roman Bold Condensed
\slshape      LM Roman Bold Oblique Condensed
```

\rmfamily correspond à \fontfamily{lmr},
\bfseries correspond à \fontseries{bx}.

.....

La face cachée des polices LATIN MODERN (suite)

La face cachée des polices LATIN MODERN (suite)

.....

La famille Sans de Latin Modern

```
\sffamily
\mdseries
\upshape      LM Sans
\slshape      LM Sans Oblique
\bfseries
\upshape      LM Sans Bold
\slshape      LM Sans Bold Oblique
\fontseries{sb}
\upshape      LM Sans Demi Condensed
\slshape      LM Sans Demi Condensed Oblique
```

\sffamily correspond à \fontfamily{lms},
\bfseries correspond à \fontseries{b}.

La famille Typewriter de Latin Modern

```
\ttfamily
\mdseries
\upshape      LM Typewriter
\itshape      LM Typewriter Italic
\slshape      LM Typewriter Oblique
\scshape      LM TYPEWRITER SMALL CAPS
\fontshape{scs1} LM TYPEWRITER SMALL CAPS OBLIQUE
\bfseries
\upshape      LM Typewriter Dark
\slshape      LM Typewriter Dark Oblique
\fontseries{1}
\upshape      LM Typewriter Light
\slshape      LM Typewriter Light Oblique
\fontseries{1c}
\upshape      LM Typewriter Light Condensed
\slshape      LM Typewriter Light Condensed Oblique
```

\ttfamily correspond à \fontfamily{lmtt},
\bfseries correspond à \fontseries{b}.

La famille Sans Extended de Latin Modern

```
\fontfamily{lmsq}
\mdseries
\upshape      LM Sans Extended
\slshape      LM Sans Extended Oblique
\bfseries
\upshape      LM Sans Extended Bold
\slshape      LM Sans Extended Bold Oblique
```

Cette police est prévues pour n'être utilisée que jusqu'à 8pt.

La famille Dunhill de Latin Modern

```
\fontfamily{lmdh}
\mdseries
\upshape      LM Sans Extended
\slshape      LM Sans Extended Oblique
```


.....
La face cachée des polices LATIN MODERN (suite)

La famille Typewriter Proportional de Latin Modern

```
\fontfamily{\lmvtt}
\mdseries
\upshape      LM Typewriter Proportional
\slshape      LM Typewriter Proportional Oblique
\bfseries
\upshape      LM Typewriter Proportional Bold
\slshape      LM Typewriter Proportional Bold Oblique
\fontseries{1}
\upshape      LM Typewriter Proportional Light
\slshape      LM Typewriter Proportional Light Oblique
```

Cette police est proportionnelle, contrairement à la *typewriter* précédente.

4.3 Autres polices d'écriture et symboles divers

.....
L'installation d'une autre famille

L^AT_EX utilise sa propre démarche pour gérer les polices d'écriture.

Il n'est donc pas facile d'installer soi-même une "jolie police trouvée sur l'ordinateur"...

Une installation T_EX récente comporte déjà un certain choix de polices, qu'il est souvent possible d'activer en utilisant un *package* spécifique.

Cela permet de s'échapper des polices par défaut COMPUTER MODERN/CM-SUPER.

Si une police existe sans *package*, on peut installer cette police globalement en redéfinissant l'une des commandes `\rmdefault`, `\sfdefault` ou `\ttdefault` :

`\renewcommand{\rmdefault}{ppl}` installe PALATINO comme police romane.

Si on souhaite seulement utiliser cette police dans une commande personnelle, on peut définir cette commande en utilisant `\fontfamily{\fmy}\selectfont` où *fmy* est le nom (L^AT_EX) de la police.

⚠ Certaines polices ne comportent pas de symboles mathématiques associés.

Dans ce cas on doit se contenter de changer la police de texte et d'utiliser les symboles mathématiques par défaut, en espérant que l'un et l'autre se marient bien.

On peut choisir une police en consultant les sites :

- The L^AT_EX Font Catalogue ⁸⁶
- A Survey of Free Math Fonts for T_EX and L^AT_EX ⁸⁷

.....
Exemples d'autres polices

Exemples de polices avec l'éventuel *package* qui l'active :

LATIN MODERN (**lmodern**) : Il se nommait Javert, et il était de la police.

LIBERTINE (**libertine**) : Il se nommait Javert, et il était de la police.

BIOLINUM (**libertine**) : Il se nommait Javert, et il était de la police.

UTOPIA (**fourier**) : Il se nommait Javert, et il était de la police.

AREV SANS (**arev**) : Il se nommait Javert, et il était de la police.

KP-FONTS (**kpfonts**) : Il se nommait Javert, et il était de la police.

CONCRETE (**ccfonts**) : Il se nommait Javert, et il était de la police.

CHARTERBT (**charter**) : Il se nommait Javert, et il était de la police.

LUCIDA : Il se nommait Javert, et il était de la police.

GARAMOND : Il se nommait Javert, et il était de la police.

MINIONPRO : Il se nommait Javert, et il était de la police.

COMICSANS : Il se nommait Javert, et il était de la police.

MARKERFELT : Il se nommait Javert, et il était de la police.

URW GROTESK : Il se nommait Javert, et il était de la police.

LUCIDABLACKLETTER : Il se nommait Javert, et il était de la police.

LUCIDAHANDWRITING : Il se nommait Javert, et il était de la police.

AIMÉE : Il se nommait Javert, et il était de la police.

TYPOAMERICAN : Il se nommait Javert, et il était de la police.

VILLAGE : Il se nommait Javert, et il était de la police. ⁸⁸

Exemples d'autres polices (suite)

Polices du projet TEX GYRE qui clone et complète les polices standard de POSTSCRIPT :
TERMES (tgtermes) : Il se nommait Javert, et il était de la police.
TIMES (mathptmx) : Il se nommait Javert, et il était de la police.
PAGELLA (tgpagella) : Il se nommait Javert, et il était de la police.
PALATINO (mathpazo) : Il se nommait Javert, et il était de la police.
BONUM (tgbonum) : Il se nommait Javert, et il était de la police.
BOOKMAN-LIGHT (bookman) : Il se nommait Javert, et il était de la police.
SCHOLA (tgschola) : Il se nommait Javert, et il était de la police.
NEWCENTURYSCHLBK (newcent) : Il se nommait Javert, et il était de la police.
ADVENTOR (tgadventor) : Il se nommait Javert, et il était de la police.
AVANT GARDE (avant) : Il se nommait Javert, et il était de la police.
HEROS (tgheros) : Il se nommait Javert, et il était de la police.
HELVETICA (helvet) : Il se nommait Javert, et il était de la police.
CURSOR (tgcursor) : Il se nommait Javert, et il était de la police.
COURIER (courier) : Il se nommait Javert, et il était de la police.
CHORUS (tgchorus) : Il se nommait Javert, et il était de la police.
ZAPFCHANCERY (chancery) : Il se nommait Javert, et il était de la police.
Les versions OPENTYPE sont disponibles.

Accès à des symboles divers

LATEX met à notre disposition des polices d'écriture et des symboles mathématiques.
Il existe aussi de nombreux symboles qui ne sont pas des symboles mathématiques.
On a vu que le package textcomp apportait déjà un certain nombres de symboles.
Le fichier Comprehensive LATEX symbol list montre comment accéder à 5 913 symboles de
texte et de mathématique (version du 9 novembre 2009) en utilisant différents packages.
Le package marvosym permet d'accéder aux symboles suivants et à bien d'autres :

Table with 4 columns showing LaTeX commands and their corresponding symbols: \Letter, \Mobilefone, \Telefon, \Email, \Coffeecup, \Football, \Bicycle, \Info, \Checkedbox, \Crossedbox, \Radioactivity, \Mundus.

D'autres packages donnent accès à d'autres symboles : pifont (voir plus loin), dingbat, bbding,
fourier, MnSymbol, ifsym, adforn...
On peut aussi accéder à des langues particulières :

Table with 2 columns showing language-specific symbols and their LaTeX commands: hierogl, phaiost, protosem, cypriot, linearA, linearb, tipa.

Accès à des symboles divers : le package pi font

Le package pi font permet d'accéder aux caractères de la police ZAPF DINGBATS par la commande
\ding{nbr} où nbr est le numéro dans le tableau ci-dessous :

Table with 20 columns and 25 rows showing various symbols and their corresponding numbers for the \ding command.

Accès à des symboles divers : le package pi font (suite)

La commande \Pisymbol{fmy}{nbr} de pi font permet d'accéder au caractère numéro nbr de la
police de nom fmy :

Table with 2 columns showing LaTeX commands for \Pisymbol and their corresponding symbols: \Pisymbol{lba}, \Pisymbol{lhc}, \Pisymbol{lsi}, \Pisymbol{lof}, \Pisymbol{zlh}, \Pisymbol{lz1}, \Pisymbol{lz2}, \Pisymbol{zsd}, \Pisymbol{ldi}.

Ces polices ne sont pas installées par défaut.
pifont permet aussi de créer des listes en puisant dans les symboles :

Code snippets for creating symbol lists using \begin{dinglist}, \end{dinglist}, \begin{dingautolist}, and \end{dingautolist}.

4.4 Les commandes de police en mathématique

Commandes générales

Dans le mode mathématique, on dispose des commandes suivantes pour changer de style :

Commande	Description	Exemple
<code>\mathnormal{-}</code>	défaut	$abcXYZ$
<code>\mathrm{-}</code>	romane	$abcXYZ$
<code>\mathbf{-}</code>	romane épaisse	$abcXYZ$
<code>\mathsf{-}</code>	sans serif	$abcXYZ$
<code>\mathit{-}</code>	italique	$abcXYZ$
<code>\mathhtt{-}</code>	typewriter	$abcXYZ$
<code>\mathcal{-}</code>	calligraphique	\mathcal{XYZ}
<code>\mathscr{-}</code>	calligraphique	\mathscr{XYZ}
<code>\mathbb{-}</code>	black board	\mathbb{XYZ}
<code>\mathfrak{-}</code>	gothique	\mathfrak{XYZ}

`\mathit{-}` active l'italique du texte, qui pourrait être différent de `\mathnormal{-}`.

`\mathscr{-}` est définie avec l'option `mathscr` du package **euscript**.

`\mathbb{-}` et `\mathfrak{-}` sont définies par le package **amsfonts**.

Les arguments de ces commandes sont en mode mathématique :

`\mathrm{E=mc^2}`, `\mathbf{E=mc^2}`, `\mathfrak{E=mc^2}`
 $E = mc^2$, $E = mc^2$, $\mathfrak{E} = mc^2$

Du texte à l'intérieur des mathématiques

⚠ `\mathrm{-}` ne sert pas à placer du texte ordinaire dans le mode mathématique.

LaTeX fournit la commande `\mbox{-}`,

le package **amsmath** fournit la commande plus évoluée `\text{-}`.

- `\mathrm{-}` reste en mode mathématique, `\mbox{-}` et `\text{-}` le quittent :
`\mathrm{E=mc^2}` ($E = mc^2$) possible, `\text{E=mc^2}` impossible,
`\mathrm{\$E=mc^2\$}` impossible, `\text{\$E=mc^2\$}` ($E = mc^2$) possible.

- `\mathrm{-}` a un comportement erratique avec les accents :
dans l'encodage OT1 avec COMPUTER MODERN ➔ erreur de compilation,
dans l'encodage T1 avec CM-SUPER ➔ warnings et absence des lettres.
- `\mathrm{-}` compose le texte dans la fonte romane des mathématiques.
Cette fonte peut-être différente de la fonte utilisée dans le texte environnant.
`\text{-}` reprend la fonte actuellement utilisée dans le texte environnant.

`\sffamily\bfseries Diff \$=\text{Diff}\neq\mathrm{Diff}
Diff = Diff \neq Diff`

`\ttfamily\itshape Diff \$=\text{Diff}\neq\mathrm{Diff}
Diff = Diff \neq Diff`

Du texte à l'intérieur des mathématiques (suite)

Les commandes `\text{-}` et `\mbox{-}` préservent (presque) les espaces :

`\text{0\neq 1}` est OK, `\mbox{0\neq 1}` est OK.
 $0 \neq 1$ est OK, $0 \neq 1$ est OK.

`\text{ind}^{\text{exp}} \neq \mbox{ind}^{\text{exp}}`
 $A_{\text{ind}}^{\text{exp}} \neq A_{\text{ind}}^{\text{exp}}$

➔ les indices et les exposants n'ont pas la bonne taille avec `\mbox{-}`.

C'est le défaut essentiel de `\mbox{-}` qui suggère d'utiliser `\text{-}`.

`\text{-}` change d'aspect en fonction du texte environnant.

On peut contraindre son contenu à ne pas subir certains changements :

`\sffamily\bfseries\itshape Diff \$\neq\text{\textup{Diff}}`
Diff \neq Diff

Le contenu de cette combinaison change de famille et de graisse (utile pour des titres par exemple) mais ne change pas de forme (pratique dans des environnements de type théorèmes qui composent le texte en italique).

La graisse épaisse en mode mathématique

⚠ `\mathbf{-}` ne sert pas à mettre en graisse épaisse des formules mathématiques :

- Elle change la forme des caractères : b (\mathbf{b}) est dans la forme italique alors que \mathbf{b} ($\mathbf{\mathbf{b}}$) est dans la forme droite.
- Elle ne donne pas accès aux symboles mathématiques en graisse épaisse :
`\mathbf{\alpha + b}` donne $\alpha + \mathbf{b}$.

On peut utiliser la commande `\boldsymbol{-}` du package **amsmath** :

`\boldsymbol{b+A\infty\mathfrak{X}\Gamma\mathbb{N}}`
 $\mathbf{b} + A\infty\mathfrak{X}\Gamma\mathbb{N} \neq \mathbf{b} + A\infty\mathfrak{X}\Gamma\mathbb{N}$

Si le symbole n'a pas de version épaisse définie (\mathbb{N} ci-dessus), on peut utiliser `\pmb{-}` (*poor man's bold*) définie par **amsmath** :

`\boldsymbol{b+A\infty\mathfrak{X}\Gamma\mathbb{N}}`
 $\mathbf{b} + A\infty\mathfrak{X}\Gamma\mathbb{N} \neq \mathbf{b} + A\infty\mathfrak{X}\Gamma\mathbb{N}$

Le résultat avec `\pmb{-}` est moins esthétique qu'avec `\boldsymbol{-}`.

Ces commandes requièrent de modifier le code mathématique.

➔ À n'utiliser que pour des symboles particuliers.

.....

La graisse épaisse en mode mathématique (suite)

La commande `\mathversion{bold}` active la graisse épaisse en mathématique sans avoir à modifier le code mathématique.

Il faut l'utiliser hors du mode mathématique.

Son contraire est `\mathversion{normal}`.

```
\mathversion{bold} $\mathfrak{X} = \alpha + b = \Gamma \div D$,
\mathversion{normal} $\mathfrak{X} = \alpha + b = \Gamma \div D$
 $\mathfrak{X} = \alpha + b = \Gamma \div D$ ,  $\mathfrak{X} = \alpha + b = \Gamma \div D$ 
```

La graisse épaisse activée par `\mathversion{bold}` est la même que celle utilisée par `\boldsymbol{-}`

→ certains symboles n'ont pas nécessairement une version épaisse définie.

Exemple : `\mathversion{bold}\mathbb{R}` donne \mathbb{R} .

`\bfseries\mathversion{bold}` passe tout en gras :

```
\bfseries\mathversion{bold}
La relation  $e^{i\pi} + 1 = 0$  est au cœur des mathématiques.
La relation  $e^{i\pi} + 1 = 0$  est au cœur des mathématiques.
```

→ utiliser l'association `\bfseries\mathversion{bold}` pour créer des titres.

.....

La taille des caractères en mode mathématique

La taille des caractères en mode mathématique est celle du texte environnant.

```
{\large large $a=b$, normalsize $a=b$, {\small small $a=b$}.
large  $a = b$ , normalsize  $a = b$ , small  $a = b$ .
```

Cependant, certains symboles peuvent apparaître plus petit en fonction du contexte, et la place des indices et des exposants peut changer :

\sum_a^b correspond au mode mathématique dans le texte ($\$ \dots \$$),

\sum_a^b correspond au mode mathématique hors du texte (equation, align...).

On peut forcer le style en utilisant les commandes `\textstyle` et `\displaystyle` :

```
{\textstyle \lim_{n \rightarrow \infty} a_n = a}
{\displaystyle \lim_{n \rightarrow \infty} a_n = a}
 $\lim_{n \rightarrow \infty} a_n = a$        $\lim_{n \rightarrow \infty} a_n = a$ 
```

→ les entrées d'un tableau `\begin{array} \dots \end{array}` sont en `\textstyle` par défaut : on peut placer `\displaystyle` dans chaque cellule pour changer ce comportement.

.....

Les polices du mode mathématique : bilan

Le fichier **Comprehensive L^AT_EX symbol list**³ montre comment accéder à de nombreux symboles mathématiques non définis par défaut.

Concernant le choix des polices mathématiques, la situation est à la fois plus compliquée et plus simple :

- plus compliquée, car installer un ensemble de polices mathématiques est beaucoup plus subtil que d'activer une police de texte ;
- plus simple, car comme ce n'est pas à la portée de n'importe quel utilisateur, on se contentera d'utiliser des *packages* qui font ça très bien...

Le problème est qu'il y a peu de *packages* qui activent d'autres polices de symboles :

- **eulervm** : il charge des polices mathématiques basées sur les polices EULER et COMPUTER MODERN. Il laisse le choix de la police de texte.
- **MnSymbol** : il charge des polices mathématiques en harmonie avec la police MINIONPRO. Mais il peut être utilisé avec d'autres polices de texte.
- **mathdesign**, **fourier**, **mathpazo**, **mathptmx**, **kpfonts**, **arev**... : ces *packages* chargent des polices de mathématiques en même temps que les polices de texte.

Cours 5 – Les mathématiques



5.1 Généralités 80
Où l'on feuillette les règles générales de la composition des mathématiques et où l'on s'initie à quelques subtilités...

5.2 Les packages **amsmath** et **mathtools** 84
Où l'on pénètre, grâce à moult exemples, dans le vaste paysage des commandes mathématiques fournies par deux modules épatants...

5.3 Les packages d'environnements de type théorèmes 97
Où l'on apprend à instruire le lecteur de la nature des énoncés mathématiques que l'on formule...

5.4 Quelques packages en vrac 99
Où l'on découvre de nouvelles possibilités dans la présentation de nos rêveries mathématiques...

5.1 Généralités

Généralités sur les mathématiques en L^AT_EX

- T_EX est l'un des premiers traitements de texte à intégrer un "mode mathématique" pour composer les formules de mathématique.
En T_EX pur, ce mode est activé par des signes \$ simples ou doubles.
Avec L^AT_EX, des environnements plus divers sont proposés.
- Deux modes sont possibles : formules dans le texte ou formules isolées au milieu de la ligne (avec ou sans numérotations).
Dans le premier cas, T_EX s'efforce de composer la formule pour qu'elle prenne le moins de place possible : comparer $\int_a^b f(x)dx = \lim_{n \rightarrow \infty} a_n$ avec

$$\int_a^b f(x)dx = \lim_{n \rightarrow \infty} a_n$$

- La première formule est composée en *textstyle* alors que la seconde est composée en *displaystyle*.
- En mode mathématique, la typographie du texte change car L^AT_EX utilise une police de "texte" spécifique : avec la police LATIN MODERN, on obtient de l'italique et des espacements fixes (famille 1mm dans l'encodage OML).
→ ça permet de distinguer a (\$a\$) de "a", mais ça produit (avec LATIN MODERN) l'affreux *Diff* plutôt que le plus acceptable *Diff* (utiliser `\text{Diff}` → *Diff*).

Généralités sur les mathématiques en L^AT_EX (suite)

- T_EX définit les contextes mathématiques \$...\$ (texte) et \$\$...\$\$ (hors texte).
L^AT_EX définit \(...\) (texte), les environnements `displaymath` (hors texte, non numéroté, synonyme : `\[...\]`) et `equation` (hors texte, numéroté).
- Les accents sont interdits en mode mathématique (cf. problèmes avec `\mathrm{-}`).
Des commandes spécifiques sont fournies : `\acute{-}` (*á*), `\bar{-}` (*ā*), `\breve{-}` (*ă*), `\check{-}` (*č*), `\ddot{-}` (*â*), `\dot{-}` (*á*), `\grave{-}` (*à*), `\hat{-}` (*â*), `\tilde{-}` (*ã*), `\widetilde{-}` (*ā*), `\widehat{-}` (*â*), `\vec{-}` (*ā*).
- Quelques commandes et environnements du mode mathématique :
 - exposants et indices : `a^2` → *a*², `a_2` → *a*₂, `a^2_n` → *a*²_n;
 - lettres grecques : `\alpha` `\Gamma` `\pi` → *α* *Γ* *π*;
 - macros d'opérateurs : `\oplus` `\cdot` `\otimes` → *⊕* *⋅* *⊗*,
 - `\cos(x) + i \sin(x) = \exp(ix)` → *cos(x) + i sin(x) = exp(ix)*;
 - symboles variés : `\infty` `\vdots` `\hbar` `\ldots` `\nabla` `\ddots` `\imath` →
∞ *⋮* *ħ* *...* *∇* *⋮* *i*;
 - éléments de formules : `\sqrt[3]{8}` = `\frac{4}{2}` → *³√8 = ⁴/₂*;
 - `\int_a^b f(x) dx = \sum_{n \in \mathbb{Z}} c_n` → *$\int_a^b f(x)dx = \sum_{n \in \mathbb{Z}} c_n$* ;
 - environnements divers : `array` pour tableaux mathématiques et les matrices...

Le chapitre 3 de **Comprehensive L^AT_EX symbol list**^{π3} donne une liste très complète de tous les symboles mathématiques utilisables avec L^AT_EX.

Les différents types de symboles en mathématique

Les symboles mathématiques sont des glyphs à part entière contenus dans des fontes spécifiques (avec un encodage *très* particulier).

D'un point de vue typographique, ces symboles sont classés en 8 classes :

1. Les symboles ordinaires, de type `\mathord` : $\alpha \infty \hbar \exists \forall \vdash \aleph \dots$
2. Les grands opérateurs, de type `\mathop` : $\sum \prod \int \oplus \cup \cap \vee \amalg \dots$
3. Les opérations binaires, de type `\mathbin` : $+ \times \oplus \div \pm \wedge \dots$
4. Les relations, de type `\mathrel` : $= \neq < \geq \nlessgtr \subset \supset \mapsto \vdash \dots$
5. Les symboles ouvrants, de type `\mathopen` : $([\{ \langle \| \dots$
6. Les symboles fermants, de type `\mathclose` : $)] \} \rangle \| \dots$
7. Les symboles de ponctuation, de type `\mathpunct` : $\dots ; \dots$
8. Les lettres, de type `\mathalpha` : $a A 1 \dots$

L'intérêt est que l'espacement avant et après un symbole est différent selon sa classe.

Seules les symboles `\mathalpha` sont affectés par des changements de style de texte.

Les symboles de type `\mathop` admettent des tailles et des positions d'indices et d'exposant différentes selon le mode mathématique :

`\oplus_{a}^{b}` `\bigoplus_{a}^{b}`

(*displaystyle*) \oplus_a^b ou (*textstyle*) \oplus_a^b

Changer la classe d'un symbole

Il peut être utile de changer la classe d'un symbole pour lui donner une signification mathématique différente.

Les commandes `\mathord{-}`... changent la classe d'un symbole, et donc les espacements :

<code>\$a \mathord{+} b\$</code>	<i>a+b</i>
<code>\$a \mathalpha{+} b\$</code>	<i>a+b</i>
<code>\$a \mathopen{+} b\$</code>	<i>a+b</i>
<code>\$a \mathclose{+} b\$</code>	<i>a+b</i>
<code>\$a \mathpunct{+} b\$</code>	<i>a+b</i>
<code>\$a \mathop{+} b\$</code>	<i>a+b</i>
<code>\$a + b\$</code>	<i>a+b</i>
<code>\$a \mathbin{+} b\$</code>	<i>a+b</i>
<code>\$a \mathrel{+} b\$</code>	<i>a+b</i>

On peut ainsi préférer $g \cdot v$ (`$g \mathord{\cdot} v$`) à $g \cdot v$ (`$g \mathalpha{\cdot} v$`) pour exprimer un produit ou l'action d'un élément g d'un groupe sur un élément v d'un espace vectoriel.

De même, on peut noter un produit vectoriel " $\vec{v} \times \vec{w}$ " de cette façon : `\vec{v} \times \vec{w}` (`$\vec{v} \mathord{\times} \vec{w}$`).

..... Symboles extensibles verticalement

Les symboles suivants sont extensibles à volonté verticalement :

(())	/ /	\uparrow ↑
[, \lbrack [], \rbrack]	\backslash \	\downarrow ↓
\{, \lbrace {	\}, \rbrace }	, \vert	\Uparrow ↑
\langle <	\rangle >	\Vert	\Downarrow ↓
\lfloor ⌊	\rfloor ⌋		\updownarrow ⇕
\lceil ⌈	\rceil ⌉		\Updownarrow ⇕
\lvert†	\rvert†		
\lVert†	\rVert†		

(† définis par le package **amsmath**)

L'agrandissement peut être géré manuellement ou automatiquement :

$\big]$	$\Big]$	$\bigg]$	$\Bigg]$
$\left\langle \sum_n \right\rangle$	$\left\langle \prod_p \right\rangle$		

Des *packages* peuvent définir d'autres symboles extensibles verticalement.

..... Symboles extensibles horizontalement

Les symboles suivants sont extensibles (presque) à volonté horizontalement :

\widetilde{-} \widetilde{abcde}	\widehat{-} \widehat{abcde}
\overline{-} \overline{abcde}	\underline{-} \underline{abcde}
\overbrace{-} \overbrace{abcde}	\underbrace{-} \underbrace{abcde}
\overbracket[-]{-}† $\overbracket[-]{abcde}$	\underbracket[-]{-}† $\underbracket[-]{abcde}$
\overleftarrow{-} \overleftarrow{abcde}	\underleftarrow{-}† \underleftarrow{abcde}
\overrightarrow{-} \overrightarrow{abcde}	\underrightarrow{-}† \underrightarrow{abcde}
\overleftrightharrow{-}† $\overleftrightharrow{abcde}$	\underleftrightharrow{-}† $\underleftrightharrow{abcde}$

(† définis par **amsmath** et ‡ définis par **mathtools**)

\overbrace{-}, \underbrace{-}, \overbracket[-]{-} et \underbracket[-]{-} acceptent des exposants et indices.

L'option de \overbracket[-]{-} et \underbracket[-]{-} fixe l'épaisseur du trait :

$\underbrace{a+b}_{\text{somme}}$	$\overbrace{a+b}^{\text{somme}}$
$\overbracket{a+b}^{\text{somme}}$	$\underbracket[5pt]{c-d}_{\text{diff.}}$

..... Commandes diverses

– Le mode mathématique définit un certain nombre d'espaces :

\thinspace, \,	⇒ ⇐	\negthinspace, \!	⇒ ⇐
\medspace, \:	⇒ ⇐	\negmedspace	⇒ ⇐
\thickspace, \;	⇒ ⇐	\negthickspace	⇒ ⇐
\enskip	⇒ ⇐		
\quad	⇒ ⇐		
\qquad	⇒ ⇐		

– La commande \not place une négation sur les symboles de type \mathrel :

$\not<$ $\not\equiv$ $\not\leq$ $\not\subseteq$ $\Rightarrow \not\neq \not\subseteq$

– Les commandes \limits et \nolimits changent le comportement des limites des sommes et intégrales :

$$\int_a^b, \sum_a^b \Rightarrow \int_a^b, \sum_a^b$$

$$\int_a^b, \sum_a^b \Rightarrow \int_a^b, \sum_a^b$$

5.2 Les packages amsmath et mathtools

Les packages amsmath et mathtools en résumé

Le package **amsmath** propose de nombreux ajouts pour composer des mathématiques.

Le package **mathtools** corrige des *bugs* dans **amsmath** et le complète.

- Environnements pour gérer la disposition des formules en *displaystyle* : `equation`, `equation*`, `gather`, `gather*`, `multline`, `multline*`, `align`, `align*`, `alignat`, `alignat*`, `split`, `multlined`, `gathered`, `aligned`, `cases`...
- Commandes et environnements pour gérer les *tags* et les *labels* des formules : `\tag{-}`, `\tag*{-}`, `\notag`, `\raisetag{-}`, `\eqref{-}`, `\refeq{-}`...
- Environnements et commandes pour composer des matrices, des fractions, des racines, des flèches extensibles...
- Commandes pour créer des opérateurs de type `sin`, `lim`...
- Commandes diverses et générales : `\substack{-}`, `\mathclap{-}`, `\intertext{-}`, `\shortintertext{-}`, `\numberwithin{-}{-}`, `\displaybreak[-]`...

Ce qui suit n'est qu'un aperçu des fonctionnalités de ces *packages*.

Voir leur documentation pour plus de renseignements et de précisions.

Dans ce qui suit, les commandes spécifiques à **mathtools** sont marquées d'un ‡, par défaut les autres commandes viennent de **amsmath**.

Le package **breqn** vaut aussi le détour : découpe automatique des longues formules, environnements de regroupements de formules, environnements de formules alignées...

L'environnement equation

`equation` sert à composer une simple équation avec ou sans numéro.

Règle : sans étoile ➔ numérotation, avec étoile ➔ pas de numéro.

Noter la différence entre `\eqref{-}` et `\ref{-}`.

```
\begin{equation}
  A+B=C \label{eq:a}
\end{equation}
Label : \eqref{eq:a}, \ref{eq:a}. Label : (1), 1.
```

```
\begin{equation}
  A+B=C \label{eq:b}\tag*{[**]}
\end{equation}
Label : \eqref{eq:b}, \ref{eq:b}. Label : ([**]), [**].
```

`\tag{-}` produit des parenthèses mais pas `\tag*{-}` (➔ problème avec `\eqref{-}`).

```
\begin{equation*}
  A+B=C
\end{equation*}
```

L'environnement `equation*` remplace la commande \TeX `$$... $$`, la commande \LaTeX `\[... \]` et l'environnement `displaymath`.

L'environnement gather

`gather` permet de rassembler plusieurs équations, sans alignements.

Chacune est numérotée, sauf si `\nonumber` ou `\notag` est utilisé.

La commande `\intertext{-}` permet d'interrompre la série de formule pour placer du texte sans quitter l'environnement `gather`. Cette commande fonctionne aussi avec `align`.

```
\begin{gather}
  A+B=B \label{gat:a} \\
  C=D+E \nonumber \\
  E=F \label{gat:c} \\
\intertext{et encore}
  G+H=I+J \notag
\end{gather}
\eqref{gat:a} et \eqref{gat:c}. (2) et (3).
```

La commande `\shortintertext{}`‡ produit moins d'espaces verticaux que `\intertext`

L'environnement align

`align` rassemble plusieurs équations, avec alignements verticaux sur les caractères `&`.

```
\begin{align}
  A+B &= B+A \label{al:a} \\
  C &= D+E \label{al:b} \\
\shortintertext{et}
  E &= F \label{al:c} \\
\end{align}
\eqref{al:a} et \eqref{al:c}. (4) et (6).
```

L'alignement peut s'effectuer sur plusieurs équations en ligne :

```
\begin{align*}
  A+B &= C & B &= C-A & A &= C-B \\
  D &= E+F & D-E &= F & D-F &= E \\
\end{align*}
```

Un coup sur deux, le rôle des `&` est différent : alignement ou séparation des colonnes.

align *versus* eqnarray

L'environnement d'alignement de formules eqnarray ne respecte pas l'espacement correct autour du signe = et positionne mal le numéro des formules trop longues :

```
\begin{equation*}
A+B = B+A
\end{equation*}
\begin{align*}
A+B &= B+A
\end{align*}
\begin{eqnarray*}
A+B &=& B+A
\end{eqnarray*}
```

```
\begin{align}
A &=& C+D+E+F+E+F+G+H
\end{align}
\begin{eqnarray}
A &=& C+D+E+F+E+F+G
\end{eqnarray}
```

→ Bannir l'environnement eqnarray au profit de align (ou split).

L'environnement multline

multline permet de présenter une équation sur plusieurs lignes : la première ligne est collée à gauche, les suivantes sont centrées, la dernière est collée à droite.

Un seul numéro est attribué à l'équation.

```
\begin{multline}
A+B-C+D-E+F = \quad \quad A+B-C+D-E+F = \\
-G+H+I-J+K-L \quad \quad \quad -G+H+I-J+K-L \\
+M-N+O-P+Q-R+S-T \quad \quad \quad +M-N+O-P+Q-R+S-T \\
+U-V+W-X+Y-Z \quad \quad \quad +U-V+W-X+Y-Z
\end{multline}
```

On peut forcer la mise à droite ou à gauche d'une ligne avec les commandes \shoveright{-} et \shoveleft{-}.

```
\begin{multline*}
A+B-C+D-E+F = \quad \quad A+B-C+D-E+F = \\
\shoveright{-G+H+I-J} \quad \quad \quad -G+H+I-J \\
\shoveleft{+K-L+M-N} \quad \quad \quad +K-L+M-N \\
+O-P+Q-R+S-T \quad \quad \quad +O-P+Q-R+S-T
\end{multline*}
```

L'environnement split

split permet de présenter une équation sur plusieurs lignes avec alignements verticaux sur les caractères &.

Cet environnement s'insère à l'intérieur d'autres environnements, equation, gather ou align, qui s'occupent alors de la numérotation.

```
\begin{equation}
\label{sp:x}
\begin{split}
A &=& B+C+F \quad \quad \quad A = B + C + F \\
&=& G \quad \quad \quad = G
\end{split}
\end{equation}
\label{ : \eqref{sp:x}}
```

```
\begin{gather}
\begin{split}
A &=& B+C+F \quad \quad \quad A = B + C + F \\
&=& G \quad \quad \quad = G
\end{split}
\end{gather}
\begin{split}
H + I = J \quad \quad \quad H + I = J
\end{split}
```

Les environnements gathered et multlined

gathered et multlined[‡] sont comme les environnements gather et multline, mais utilisés comme sous blocs d'une équation.

Ces environnements acceptent une option pour le positionnement vertical (t,c,b).

```
\begin{equation}
a < 0 \Rightarrow \begin{gathered}
A = B + C \\
D = E
\end{gathered}
\end{equation}
```

```
\begin{equation}
a < 0 \Rightarrow \begin{multlined}[t]
A+B = C+D+E+F+G+H \quad \quad \quad A+B = C+D+E+F+G+H \\
+I+J+K+L+M+N \quad \quad \quad +I+J+K+L+M+N \\
+O+P+Q+R+S \quad \quad \quad +O+P+Q+R+S
\end{multlined}
\end{equation}
```

multlined[‡] permet des ajustements fins sur la largeur du bloc.

Les environnements aligned et alignat

aligned et alignat sont des variantes de align.

aligned est la version “sous bloc” de align.

```
\begin{equation}
\begin{aligned}[b]
A &= B \quad \quad \quad A = B \\
&= C \quad \quad \quad = C \Rightarrow a < 0
\end{aligned}
\end{equation}
```

(15)

alignat permet de maîtriser l'espacement entre les colonnes.

```
\begin{alignat}{3}
A+B &= C & & B &= C-A & A &= C-B \\
D &= E+F & \quad & D-E &= F & & D-F &= E
\end{alignat}
```

$$A + B = C \quad B = C - A \quad A = C - B \quad (16)$$

$$D = E + F \quad D - E = F \quad D - F = E \quad (17)$$

Le \quad écarte la première et la seconde colonne, alors que la dernière colonne est collée à la seconde. L'argument 3 fait référence au nombre de colonnes.

Les environnements de type cases

cases est une variante de aligned pour gérer des “cas” :

```
\begin{equation}
A = \begin{cases}
B & \text{si } a < 0 \\
C & \text{si } a \geq 0
\end{cases}
\end{equation}
```

$$A = \begin{cases} B & \text{si } a < 0 \\ C & \text{si } a \geq 0 \end{cases} \quad (18)$$

mathtools définit les environnements drcases[†], dcases[†], rcases[†], rcases[†], drcases[†], drcases[†], cases[†].

dcases compose en *displaystyle*. Les versions “rcases” placent l'accolade à droite, les versions étoilées produisent du texte dans la seconde colonne :

```
\begin{equation*}
\begin{drcases*}
A = B & \text{évident} \\
2 \int_0^1 x \, dx = 1 & \text{facile}
\end{drcases*}
\end{equation*}
```

$$\left. \begin{array}{ll} A = B & \text{évident} \\ 2 \int_0^1 x \, dx = 1 & \text{facile} \end{array} \right\} \Rightarrow a > 0$$

L'environnement flalign

flalign est une variante de align qui sépare le plus possible les colonnes pour remplir la largeur de la page.

```
\begin{flalign*}
A+B &= C & & B &= C-A & A &= C-B \\
D &= E+F & & D-E &= F & & D-F &= E
\end{flalign*}
```

$$A + B = C \quad B = C - A \quad A = C - B$$

$$D = E + F \quad D - E = F \quad D - F = E$$

Version avec align* :

$$A + B = C \quad B = C - A \quad A = C - B$$

$$D = E + F \quad D - E = F \quad D - F = E$$

Version avec flalign :

$$A + B = C \quad B = C - A \quad A = C - B \quad (19)$$

$$D = E + F \quad D - E = F \quad D - F = E \quad (20)$$

Commandes associées à align et alignat

\MoveEqLeft[-][‡] décale vers la gauche une ligne dans align.

L'argument optionnel est la distance en unités em :

```
\begin{align*}
\MoveEqLeft[3] A+B+C+D+E \\
&= F+G+H+I+J+K \\
&\leq L+M+N+O+P
\end{align*}
```

$$A + B + C + D + E = F + G + H + I + J + K \leq L + M + N + O + P$$

\MoveEqLeft[-] prend la place du premier symbole &.

\ArrowBetweenLines[-][‡] insère une flèche entre des lignes d'équations :

```
\begin{alignat*}{2}
& A &= B \\
\ArrowBetweenLines[\Downarrow]
& B &= A
\end{alignat*}
```

$$\begin{array}{ll} A = B \\ \Downarrow \\ B = A \end{array}$$

.....

Alerte aux raccourcis...

Pour économiser de la frappe, il est souvent commode de créer des raccourcis du type

```
\newcommand{\beq}{\begin{equation}}
\newcommand{\eeq}{\end{equation}}
pour délimiter une équation.
```

Attention

Ces raccourcis ne peuvent pas être définis avec des environnements d'alignement, comme `a align`, `gather`, `multline`...

La raison profonde est la suivante, et elle peut se retrouver dans d'autres circonstances.

1. L'algorithme (compliqué) de mise en forme de ces types d'équations requiert deux passages sur le contenu de l'environnement.
→ T_EX doit pouvoir absorber dans son entier le contenu de l'environnement.
2. Dès qu'il rentre dans ce type d'environnement par une commande `\begin{env}`, il cherche *immédiatement* la sortie sous la forme `\end{env}`.

Si des raccourcis `\bev` et `\eev` ont été créés pour ces commandes, voici ce qui se passe :

- La commande `\bev` est interprétée par T_EX comme la commande `\begin{env}`.
→ Il cherche alors la première expression `\end{env}` qui suit...
- Or cette expression n'existe pas, puisqu'elle est "cachée" dans `\eev`.
→ T_EX parcourt tout le texte sans succès et se plaint (de façon incompréhensible!).

.....

L'environnement subequations

`subequations` agit sur la numérotation des formules qu'il contient :

```
\begin{subequations}
\begin{equation}
A=B
\end{equation}
Texte entre équations.
\begin{equation}
D=C \label{eq:mi}
\end{equation}
\begin{align}
E &= F & E' &= F' \\
G &= H & G' &= H'
\end{align}
\end{subequations}
Fin subequation.
\begin{equation}
I=J \label{eq:fi}
\end{equation}
\eqref{eq:mi} et \eqref{eq:fi}.
```

$$A = B \quad (21a)$$

Texte entre équations.

$$D = C \quad (21b)$$

$$E = F \quad E' = F' \quad (21c)$$

$$G = H \quad G' = H' \quad (21d)$$

Fin subequation.

$$I = J \quad (22)$$

(21b) et (22).

.....

Les matrices

`matrix` produit un tableau mieux espacé que `array` de L^AT_EX :

```
\begin{matrix}
a & b \\
a & b \quad \quad c & d \\
c & d
\end{matrix}
```

`pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` et `Vmatrix` produisent :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \left\{ \begin{matrix} a & b \\ c & d \end{matrix} \right\} \quad \left| \begin{matrix} a & b \\ c & d \end{matrix} \right| \quad \left\| \begin{matrix} a & b \\ c & d \end{matrix} \right\|$$

`matrix*†`, `pmatrix*†`, `bmatrix*†`, `Bmatrix*†`, `vmatrix*†` et `Vmatrix*†` permettent de contrôler l'alignement vertical :

```
\begin{Vmatrix*}[r]
-1 & 2 \quad \quad 3 & -4 \\
\end{Vmatrix*}
```

$$\begin{Vmatrix} -1 & 2 \\ 3 & -4 \end{Vmatrix}$$

```
\begin{bmatrix*}[l]
a_{\scriptscriptstyle n} & b \\
c & d_{\scriptscriptstyle m}
\end{bmatrix}
```

$$\begin{bmatrix} a_n & b \\ c & d_m \end{bmatrix}$$

.....

Les matrices (suite)

`smallmatrix` permet de créer des petites matrices pour le texte :

```
$A=\left(\begin{smallmatrix} a&b \\ c&d \end{smallmatrix}\right)$
A = \left( \begin{matrix} a & b \\ c & d \end{matrix} \right)
```

`\hdotsfor[-]{-}` permet de placer des pointillés sur plusieurs colonnes :

```
\begin{vmatrix}
a & b & c & d & e \\
f & \hdotsfor{3} & g \\
\hdotsfor{2} & \hdotsfor{5} \\
h & i & j & k & l
\end{vmatrix}
```

$$\begin{vmatrix} a & b & c & d & e \\ f & \dots & \dots & \dots & g \\ \dots & \dots & \dots & \dots & \dots \\ h & i & j & k & l \end{vmatrix}$$

L'argument optionnel sert à ajuster l'espacement horizontal entre les points (c'est un facteur multiplicatif).

Flèches extensibles

amsmath définit des flèches extensibles :

```
$A \xleftarrow{\text{vers la gauche}} B \xrightarrow[\text{en dessous}]{\text{vers la droite}} C$
```

$$A \xleftarrow{\text{vers la gauche}} B \xrightarrow[\text{en dessous}]{\text{vers la droite}} C$$

mathtools ajoute les flèches suivantes :

```
\xleftarrow[en haut]{en bas} \hookleftarrow[en haut]{en bas}
\xLeftarrow[en haut]{en bas} \hookrightarrow[en haut]{en bas}
\xrightarrow[en haut]{en bas} \xmapsto[en haut]{en bas}
\xLeftrightarrow[en haut]{en bas}
```

$$\begin{array}{ll} \xleftarrow[en haut]{en bas} & \hookleftarrow[en haut]{en bas} \\ \xLeftarrow[en haut]{en bas} & \hookrightarrow[en haut]{en bas} \\ \xrightarrow[en haut]{en bas} & \xmapsto[en haut]{en bas} \\ \xLeftrightarrow[en haut]{en bas} & \end{array}$$

Le package **extpfeil** définit les flèches extensibles suivantes :

```
\xrightarrow[en haut]{en bas} \xleftarrow[en haut]{en bas} \xleftrightarrow[en haut]{en bas} \xleftrightarrow[en haut]{en bas}
```

$$\xrightarrow[en haut]{en bas} \xleftarrow[en haut]{en bas} \xleftrightarrow[en haut]{en bas} \xleftrightarrow[en haut]{en bas}$$

Commandes diverses

Placée avant un `\, \displaybreak[num]` avec $num = 0, \dots, 4$ suggère d'interrompre à cet endroit une série d'équations pour passer à la page.

Fractions : `\frac{-}{-}` remplace la commande `\over` de \LaTeX .

`\dfrac{-}{-}` force le *displaystyle* et `\tfrac{-}{-}` force le *textstyle* :

```
$A = \frac{1}{2} = \dfrac{1}{2} = \tfrac{1}{2}$
```

$$A = \frac{1}{2} = \dfrac{1}{2} = \tfrac{1}{2}$$

Coefficients binomiaux :

```
$A = \binom{n}{p} = \dbinom{n}{p} = \tbinom{n}{p}$
```

$$A = \binom{n}{p} = \dbinom{n}{p} = \tbinom{n}{p}$$

Intégrales multiples :

```
$\displaystyle \int_X f(x) dx = \iint_Y g(y) dy = \iiint_Z h(z) dz = \dots \int_T r(t) dt$
```

$$\int_X f(x) dx = \iint_Y g(y) dy = \iiint_Z h(z) dz = \dots \int_T r(t) dt$$

Commandes diverses (suite)

Deux points verticalement alignés, `\vcentcolon`† :

```
$a \vcentcolon= b := c$ ($\vcentcolon :$)
```

$$a \vcentcolon= b := c (:) :$$

`\sideset{-}{-}` ajoute des symboles en indices et exposants à un opérateur large :

```
\sideset{a^b}{c^d} \prod \quad \quad \quad \prod_{n=0}^{\infty} \quad \quad \quad \bigoplus_c
```

$$\prod_{n=0}^{\infty} \quad \quad \quad \bigoplus_c$$

`\prescript{-}{-}`† place des indices et exposant avant l'expression qui suit :

```
$\prescript{4}{12}C^{5+}_{2} \quad \quad \quad \prescript{14}{2}C^{5+}_{2} \quad \quad \quad \prescript{14}{2}C^{5+}_{2}
```

$$\prescript{4}{12}C^{5+}_{2} \quad \quad \quad \prescript{14}{2}C^{5+}_{2} \quad \quad \quad \prescript{14}{2}C^{5+}_{2}$$

`\adjustlimits` ajuste verticalement les limites successives :

```
$\displaystyle \lim_{n \rightarrow \infty} \max_{p^2 \geq n} \quad \quad \quad \max_{n \rightarrow \infty} \lim_{p^2 \geq n}$
```

$$\lim_{n \rightarrow \infty} \max_{p^2 \geq n} \quad \quad \quad \max_{n \rightarrow \infty} \lim_{p^2 \geq n}$$

Commandes diverses (suite)

`\mathclap{-}`†, `\mathllap{-}`† et `\mathrlap{-}`† réduisent leur contenu à une largeur nulle en position centrée, gauche ou droite :

```
\sum_{0 < i < j < n} X_{ij} = \sum_{\mathclap{0 < i < j < n}} X_{ij} \quad \quad \quad \prod_{0 < s < n} \sum_{\mathllap{0 < r < n}} a_r
```

$$\sum_{0 < i < j < n} X_{ij} = \sum_{\mathclap{0 < i < j < n}} X_{ij} \quad \quad \quad \prod_{0 < s < n} \sum_{\mathllap{0 < r < n}} a_r$$

(autres utilisations possibles !)

`\smashoperator{-}`† fait de même au niveau de l'opérateur en entier.

L'argument optionnel précise le(s) côté(s) à traiter :

```
\sum_{0 < i < j < n}^{\infty} V_{ij} = \smashoperator[r]{\sum_{0 < i < j < n}^{123456}} X_{ij} = \smashoperator[l]{\prod_{0 < i < j < n}} Y_{ij} = \smashoperator[l]{\bigvee_{0 < i < j < n}} Z_{ij}
```

$$\sum_{0 < i < j < n}^{\infty} V_{ij} = \smashoperator[r]{\sum_{0 < i < j < n}^{123456}} X_{ij} = \smashoperator[l]{\prod_{0 < i < j < n}} Y_{ij} = \smashoperator[l]{\bigvee_{0 < i < j < n}} Z_{ij}$$

Superposition

`\overset{-}{-}` et `\underset{-}{-}` permettent de superposer des symboles :

```
$\overset{*}{X}\overset{\text{f}}{\rightarrow}\underset{*}{Y}$
```

$$X \xrightarrow{*} Y$$

`\substack{-}` superpose des expressions, pour des sommes par exemple.

Le résultat est de la taille des indices et exposants :

```
$\displaystyle\sum_{\substack{0 < i < m \\ 0 < j < n}} P_{i,j}
```

$$\sum_{\substack{0 < i < m \\ 0 < j < n}} P_{i,j} = \sum_{a=0}^b P_{i,j}$$

`\genfrac{-}{-}{-}{-}{-}` est une commande générale qui définit les fractions et les coefficients binomiaux et qu'on peut utiliser directement :

```
$a = \genfrac{|}{|}{0pt}{0}{A}{B} = \genfrac{.}{|}{2pt}{1}{C}{D}$
```

$$a = \frac{A}{B} = \frac{C}{D} = \left| \frac{A}{B} \right|$$

Style des labels des formules

`\numberwithin{equation}{section}` remet le compteur d'équation à 0 à chaque section et le label est du type (2.4).

`\newtagform{-}{-}{-}` et `\usetagform{-}` définissent les styles des labels.

```
\newtagform{bra}{[]{}}
\newtagform{brab}{[\textbf]{[]{}]}
\usetagform{bra}
\begin{equation}
E = mc^2 \label{eq-tf}
\end{equation}
\eqref{eq-tf}, \eqref{eq-tfb} [23], [24]
\usetagform{brab}
E \neq mc^3 \label{eq-tfb} (23), (24)
\end{equation}
\eqref{eq-tf}, \eqref{eq-tfb}\
\usetagform{default}
\eqref{eq-tf}, \eqref{eq-tfb}
```

⚠ le label peut ne pas être référencé dans le style activé au moment de l'équation.

Définir un nouvel opérateur

Les commandes de type opérateurs ont un comportement spécifique (espace, fontes) :

```
$r \sin \theta$ r sin θ
$r \text{sin} \theta$ r sin θ
$r \mathrm{sin} \theta$ r sin θ
```

La commande `\DeclareMathOperator{-}{-}` (dans le préambule) déclare un opérateur de type “sin” ou “log” : gestion des espaces avant et après, gestion de la police.

```
\DeclareMathOperator{\antisin}{antisin}
On pose $\antisin(x) = \cos(x)$ \quad
{\itshape on pose $ \antisin(x) = \cos(x)$}
On pose antisin(x) = cos(x) on pose antisin(x) = cos(x)
```

`\text{antisin}` donnerait en italique : *antisin*(y) = cos(y) ➡ pas satisfaisant.

`\DeclareMathOperator*{-}{-}` déclare un opérateur de type “lim” avec le bon comportement des “indices” :

```
\DeclareMathOperator*{\antimax}{antimax}
$\antimax_{a \in A} = \min_{a \in A} A$ \quad
$\displaystyle\antimax_{a \in A} = \min_{a \in A} A$
antimax_{a \in A} = \min_{a \in A} A
```

Les opérateurs déjà définis

<code>\arccos</code> arccos	<code>\arcsin</code> arcsin	<code>\arccos</code> arccos
<code>\arctan</code> arctan	<code>\arg</code> arg	<code>\cos</code> cos
<code>\cosh</code> cosh	<code>\cot</code> cot	<code>\coth</code> coth
<code>\csc</code> csc	<code>\deg</code> deg	<code>\det</code> det
<code>\dim</code> dim	<code>\exp</code> exp	<code>\gcd</code> gcd
<code>\hom</code> hom	<code>\inf</code> inf	<code>\injlim</code> injlim
<code>\ker</code> ker	<code>\lg</code> lg	<code>\lim</code> lim
<code>\liminf</code> liminf	<code>\limsup</code> limsup	<code>\ln</code> ln
<code>\log</code> log	<code>\max</code> max	<code>\min</code> min
<code>\Pr</code> Pr	<code>\projlim</code> projlim	<code>\sec</code> sec
<code>\sin</code> sin	<code>\sinh</code> sinh	<code>\sup</code> sup
<code>\tan</code> tan	<code>\tanh</code> tanh	<code>\varinjlim</code> \varinjlim
<code>\varliminf</code> \varliminf	<code>\varlimsup</code> \varlimsup	<code>\varprojlim</code> \varprojlim

Pas trace de la trace... ➡ définir `\DeclareMathOperator{\tr}{tr}`.

On peut ajouter à cette liste les “modulos” :

```
\mod m \equiv n \mod p
\bmod m \equiv n \mod p
\pmod m \equiv n \mod p
```

Définir des commandes avec délimiteurs

`\DeclarePairedDelimiter`[†] permet de définir des commandes personnelles de type “valeur absolue” ou “norme” :

```
\DeclarePairedDelimiter\abs{\lvert}{\rvert}
\DeclarePairedDelimiter\norm{\lVert}{\rVert}
\DeclarePairedDelimiter\delfloor{\lfloor}{\rceil}
$\abs{a} \quad \abs*{\frac{a}{b}} \quad \norm{v} \quad \delfloor{M} \quad \delfloor*{\sum_n M_n}$
 $|a| \quad \left|\frac{a}{b}\right| \quad \|v\| \quad [M] \quad [\sum_n M_n]$ 
```

La version étoilée de la commande définie ajuste correctement en hauteur les délimiteurs.

`\DeclarePairedDelimiterX`[‡] étend la commande précédente en définissant une commande avec plus d'un argument :

```
\DeclarePairedDelimiterX\innerp[2]{\langle}{\rangle}{#1,#2}
\DeclarePairedDelimiterX\bracket[3]{\langle}{\rangle}%
{#1\,,\delimsize\vert\,,#2\,,\delimsize\vert\,,#3}
$\innerp*{A}{\frac{1}{2}} \quad \bracket[\Big]{B}{\sum_k a_k}{C} \quad \bracket*{B}{\sum_k a_k}{C}$
 $\langle A, \frac{1}{2} \rangle \quad \langle B \mid \sum_k a_k \mid C \rangle \quad \langle B \mid \sum_k a_k \mid C \rangle$ 
```

5.3 Les packages d'environnements de type théorèmes

Le package `amsthm`

Le package `amsthm` améliore l'environnement `theorem` défini dans `LaTeX` en fournissant une syntaxe pour définir et utiliser des styles.

```
\newtheorem{lem}{Lemme}
```

définit un nouvel environnement `lem` dont le titre est “Lemme”.

```
\newtheorem{prop}[thm]{Proposition}
```

définit `prop` dont le compteur de numérotation est le même que celui de `thm`.

```
\newtheorem{defn}[Définition][section]
```

définit `defn` dont le compteur de numérotation est remis à 0 à chaque début de section.

```
\theoremstyle{remark}
```

```
\newtheorem{rem}{Remarque}
```

définit `rem` avec le style “remark”.

```
\newtheoremstyle{note}{2pt}{3pt}{\bfseries}%
```

```
{2cm}{\sffamily}{:}{2em}{}
```

définit un nouveau style “note” (voir la documentation pour la syntaxe).

L'environnement `proof` sert aux démonstrations.

La commande `\qedsymbol` place un symbole de fin de démonstration (personnalisable).

`amsthm` et `babel`

Il est possible de définir des environnements de type théorèmes qui sont traduits automatiquement en fonction de la langue sélectionnée avec le package `babel` :

```
\newcommand{\theoremname}{Theorem}
```

```
\addtocaptionsfrench{\renewcommand{\theoremname}{Théorème}}
```

```
\addtocaptionsenglish{\renewcommand{\theoremname}{Theorem}}
```

```
\newtheorem{thm}{\theoremname}[chapter]
```

```
\newcommand{\remarkname}{Remark}
```

```
\addtocaptionsfrench{\renewcommand{\remarkname}{Remarque}}
```

```
\addtocaptionsenglish{\renewcommand{\remarkname}{Remark}}
```

```
\newtheorem{rmk}{\remarkname}
```

Voir la documentation de `babel` pour la syntaxe de la commande `\addto`.

Le package **ntheorem**

Le package **ntheorem** étend les fonctionnalités de **amsthm** :

- Possibilités plus nombreuses pour personnaliser le style des théorèmes : entête, numérotation, corps du texte, symbole de fin...
- Compatibilité avec le package **hyperref** et gestion améliorées de références croisées (`\thref{-}`).
- Possibilité d'utiliser des commandes d'encadrement autour des théorèmes : option `framed`.
- Liste de théorèmes avec possibilité de personnalisation de la présentation : `\listtheorems{-}`, `\theoremlisttype{-}`...

```
\theoremstyle{break} % style prédéfini
\theoremnumbering{arabic} % numérotation
\theoremsymbol{\ensuremath{\blacksquare}} % symbole de fin
\theorembodyfont{\slshape} % style du texte
\theoremheaderfont{\normalfont\bfseries} % style de l'entête
\theoremseparator{\ :} % séparation de l'entête
```

```
\newtheorem{definition}{Définition}
définit definition dans le style ainsi établi.
```

Le package **thmtools**

thmtools est une surcouche à **amsthm** ou à **ntheorem** (qu'il faut appeler au préalable).

Ses commandes étendent de nombreuses possibilités : gestion plus avancée des styles, encadrement des théorèmes, compatibilité avec les packages **hyperref**, **nameref** et **cleveref**, liste des théorèmes, répétition de théorèmes déjà tapés...

```
\declaretheorem[sharecounter=theorem]{lemma}
\declaretheorem[numbered=no, style=remark]{remark}
\declaretheorem[name=Theorem, numberwithin=section,
  refname={theorem,theorems}, Refname={Theorem,Theorems}]{thm}
\declaretheorem[shaded={bgcolor=Lavender, textwidth=12em}]{BoxI}
\declaretheorem[shaded={rulecolor=Lavender,
  rulewidth=2pt, bgcolor={rgb}{1,1,1}}]{BoxII}
```

définit une série d'environnements de type théorème.

```
\declaretheoremstyle[spaceabove=6pt, spacebelow=6pt,
  headfont=\normalfont\bfseries, notefont=\mdseries,
  notebraces={{}}, bodyfont=\normalfont,
  postheadspace=1em, qed=\qedsymbol]{mystyle}
définit un nouveau style.
```

5.4 Quelques packages en vrac

Des lettres exotiques pour les mathématiques

Par défaut, L^AT_EX définit la commande

```
\mathcal{-} ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Le package **amsfonts** définit la commande suivante pour le gothique :

```
\mathfrak{-} o123456789 abcdefghijklmnopqrstuvwxyz
                A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Le package **euca1**, appelé avec l'option `mathscr`, définit la commande :

```
\mathscr{-} ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Sans l'option `mathscr`, c'est la commande `\mathcal{-}` de L^AT_EX qui est redéfinie.

Le package **mathrsfs** définit la commande :

```
\mathscr{-} A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

→ conflit de nom entre **mathrsfs** et **euca1**.

Des lettres à doubles barres

Le package **amsfonts** définit la commande :

```
\mathbb{-} ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Le package **dsfont** définit la commande :

```
\mathds{-} 1 ABCDEFGHIJKLMNOPQRSTUVWXYZl h k
```

L'option `sans` de **dsfont** produit des lettres sans sérif :

```
\mathds{-} 1 ABCDEFGHIJKLMNOPQRSTUVWXYZl h k
```

Il est possible de récupérer le **1** de cette fonte qui n'est pas dans `\mathbb{-}` en définissant la commande suivante :

```
\newcommand{\bbone}{\text{\usefont{U}{dsss}{m}{n}\char49}}
```

```
$\bbone \in A \quad \frac{\bbone}{2} \quad \sigma_{\bbone}$
1 ∈ A     $\frac{1}{2}$      $\sigma_1$ 
```

Décorations de formules

La commande `\fbox{-}` de \LaTeX peut encadrer une formule entière dans le texte ou dans un environnement de type `equation`.

La commande `\boxed{-}` de **amsmath** permet d'encadrer un morceau de formule.

```
\fbox{$E=mc^2$}\quad \boxed{E=mc^2}
```

Le package **empheq** permet plus d'options pour décorer des formules.

```
\begin{empheq}[box=\fbox]{align}
E &= h \nu \notag\\
E &= mc^2
\end{empheq}
```

$$E = hv$$

$$E = mc^2$$
(25)

```
\begin{empheq}%
[left=A \Rightarrow \empheqlbrace,
innerbox=\fbox]{align*}
B &= C\\
D &= E
\end{empheq}
```

$$A \Rightarrow \begin{cases} B = C \\ D = E \end{cases}$$

empheq reconnaît tous les environnements définis par **amsmath**.

Décorations de formules (suite)

On peut utiliser sa propre commande d'encadrement :

```
\definecolor{myblue}{rgb}{.8, .8, 1}
\newcommand\mybluebox[1]{%
\colorbox{myblue}{\hspace{1em}\#1\hspace{1em}}}
\begin{empheq}[box=\mybluebox]{gather}
A+B = C\\
D = E+F
\end{empheq}
```

$$A + B = C$$
(26)

$$D = E + F$$
(27)

On peut faire appel aux boîtes définies par le package **fancybox** :

```
\begin{empheq}[box=\ovalbox]{equation}
A+B = C+D+E
\end{empheq}
```

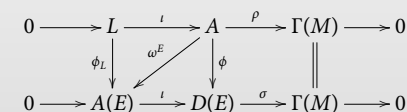
$$A + B = C + D + E$$
(28)

Consulter la documentation de **empheq** pour plus de renseignements et d'exemples.

Diagrammes commutatifs

Le package **xy**, avec les options `matrix` et `arrow`, permet de composer des diagrammes :

```
\begin{equation*}
\xymatrix{
{0} \ar[r]
& \{L\} \ar[r]^-{\iota} \ar[d]_-{\phi_L}
& \{A\} \ar[r]^-{\rho} \ar[d]^-{\phi} \ar[l]_-{\omega^E}
& \{\Gamma(M)\} \ar[r] \ar@{=}[d]
& \{0\} \\
& \{0\} \ar[r]
& \{A(E)\} \ar[r]^-{\iota}
& \{D(E)\} \ar[r]^-{\sigma}
& \{\Gamma(M)\} \ar[r]
& \{0\}
}
\end{equation*}
```



Cabinet de curiosités

Le package **12many** définit une macro pour générer un ensemble d'entiers. Plusieurs styles peuvent être utilisés et définis :

```
\setOTMstyle{dots}
$\nto{3}{4}$
```

$$\{3, \dots, 4\}$$

```
\setOTMstyle[naturals={\mathbb{N}},
var={n}]{set}
$\nto{3}{4}$
```

$$\{n \in \mathbb{N} \mid 3 \leq n \leq 4\}$$

Le package **cancel** définit des commandes pour barrer des morceaux de formules :

```
\cancel{a^2} + b + \bcancel{c_n} - \xcancel{H_I} = 0$
```

$$a^2 + b + c_n - H_I = 0$$

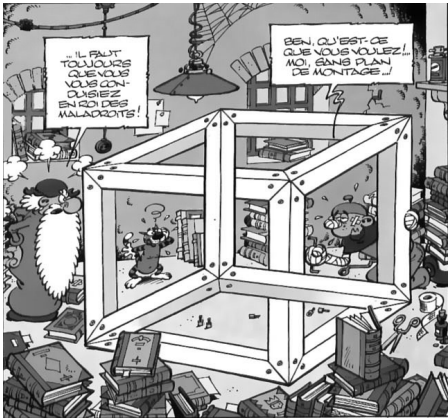
Le package **fouridx** définit une commande pour placer 4 indices à un symbole :

```
\fourIdx{12}{3}{4}{56}{\left(\frac{1}{b}\right)}\quad
\fourIdx{(k)}{n}{x}{\overline{x}}A$
```

$${}_{3\left(\frac{1}{b}\right)}^{12}_{56} {}^{(k)}_n A_{\overline{x}}$$

(Voir aussi la commande `\prescript{-}{-}` de **mathtools**.)

Cours 6 – Des packages pour sauver les apparences



6.1 Des outils techniques utiles 104
Où l'on apprend à se simplifier la vie en s'aidant de commandes bien utiles...

6.2 La gestion de la couleur 108
Où l'on apprend à colorier son document...

6.3 La gestion des tableaux 110
Où on découvre comment réaliser des tableaux avec des colonnes de différentes formes...

6.4 La gestion des listes 113
Où l'on s'initie à l'art difficile de bien faire ses listes...

6.5 La mise en page 115
Où l'on apprend à changer l'aspect de différents éléments de la page de texte...

6.6 Autres éléments du document 119
Où l'on apprend à personnaliser la page de garde et à encadrer de façon amusante du texte...

6.1 Des outils techniques utiles

La mise en boîte

Dans son principe fondamental, L^AT_EX gère des boîtes : chaque lettre est une boîte, chaque mot est un ensemble de boîtes (les lettres), les paragraphes sont des grosses boîtes...

—phénix— → 

Chaque boîte est définie par sa largeur (*width*), sa hauteur au dessus de la ligne de base (*height*) et sa profondeur en dessous de la ligne de base (*depth*).

Ainsi, lorsque L^AT_EX a terminé de composer un tableau, ce dernier devient une boîte dont il ne retient que les attributs de dimensions afin de la placer dans la page.

Dans de nombreuses commandes (comme celles qui suivent), il est possible d'utiliser les dimensions naturelles des objets L^AT_EX sur lesquelles elles s'appliquent :

- `\width` désigne la largeur,
- `\height` désigne la hauteur,
- `\depth` désigne la profondeur,
- `\totalheight` désigne la hauteur totale `\height+\depth`.

La commande `` n'imprime pas son contenu tout en créant la boîte associée :


`ab` → a b

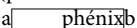
Cette commande permet de dépanner dans certaines mises en page difficiles.

La commande `\makebox`

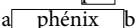
La commande `\makebox[largeur][position]{-}` crée une boîte de largeur finale *largeur* dont le contenu est positionné selon *position*=c,l,r,s.

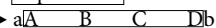
Dans les exemples suivants, un cadre est ajouté pour visualiser la boîte finale.

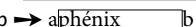
`a\makebox{phénix}b` → a  b (permet de créer des blocks insécables)

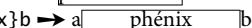
`a\makebox[5em][r]{phénix}b` → a  b

`a\makebox[1.5em][l]{phénix}bcdef` → a  bcdef

`a\makebox[5em][c]{phénix}b` → a  b


`a\makebox[8em][s]{A B C D}b` → a  b


`a\makebox[2\width][l]{phénix}b` → a  b

`a\makebox[10\height][c]{phénix}b` → a  b

`\framebox` est comme `\makebox` avec un cadre en plus.

Les longueurs `\fboxrule` et `\fboxsep` désignent l'épaisseur du trait et la séparation intérieure avec le texte.

`\framebox{phénix}` →  (défaut : `\fboxrule=0.4pt`, `\fboxsep=3.0pt`)

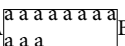
`\setlength{\fboxrule}{1.5pt}\setlength{\fboxsep}{1pt}` → 

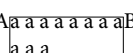
⚠ Les commandes précédentes ne peuvent accepter qu'une boîte déjà formatée (quelques mots, un tableau, une image...) mais pas un paragraphe à mettre en forme.

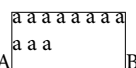
La commande `\parbox`

La commande `\parbox[position][hauteur][pos.interne]{largeur}{-}` crée une boîte dont le contenu est un paragraphe entier qu'il formate sur des lignes de largeur *largeur*.

La hauteur finale est fixée par *hauteur*, le contenu est aligné verticalement selon *pos.interne*=t,c,b,s, le paramètre *position*=t,c,b fixe l'alignement vertical de la boîte finale par rapport au texte ambiant.

`A\parbox{5em}{...}B` → A  B

`A\parbox[t][\height][t]{5em}{...}B` → A  B

`A\parbox[b][1.5\height][t]{5em}{...}B` → A  B

Il existe un environnement équivalent nommé `minipage` :

`A\begin{minipage}[c][8ex][b]{5em}`

... → A  B


Lorsqu'on souhaite utiliser des environnements pour formater le paragraphe (`center`, `flushleft`, `flushright`...) il est préférable d'utiliser `minipage` pour la gestion des espaces verticaux.


La commande `\raisebox`


La commande `\raisebox{distance}[hauteur][profondeur]{-}` relève son contenu d'une distance *distance* (+ vers le haut, - vers la bas), et produit une boîte finale de hauteur *hauteur* et de profondeur *profondeur*.


Cette commande ne peut contenir qu'une boîte déjà formatée.


`a\raisebox{1ex}{phénix}b` → a  b

`a\raisebox{1ex}[1ex][0pt]{phénix}b` → a  b


`a\raisebox{-1ex}{phénix}b` → a  b

`a\raisebox{-1ex}[2ex][12pt]{phénix}b` → a  b

`a\raisebox{-1ex}[\height][\depth]{phénix}b` → a  b

`a\raisebox{1ex}[0pt][0pt]{phénix}b` → a  b

En combinant les commandes `\makebox` et `\raisebox`, on peut produire une boîte dont toutes les dimensions sont nulles :

`a\raisebox{1.5ex}[0pt][0pt]{\makebox[0pt][c]{phénix}}b` → a  b

`a\raisebox{1.5ex}[0pt][0pt]{\makebox[0pt][l]{phénix}}b` → a  b

.....

Le *package* `calc`

Dans le langage \TeX , les opérations arithmétiques élémentaires sont réalisées grâce aux commandes `\advance` et `\multiply` dans une syntaxe assez lourde.

Le *package* `calc` apporte de la souplesse en donnant la possibilité de réaliser des opérations avec des notations mathématiques plus habituelles.

```
\newcounter{local} \setcounter{local}{2}
```

```
\setcounter{local}{\value{local}*3 + 5}
```

La commande `\the local` donne alors la valeur 11.

```
\newlength{\lgrlocal}
```

```
\setlength{\lgrlocal}{0.75\textheight-\textwidth/2}
```

est accepté pour définir une dimension.

```
a\makebox[2\width+1em][c]{phénix}b → a phénix b
```

Les opérations possibles sont : +, -, *, /, `\ratio{-}{-}` (rapport de deux longueurs) et `\real{-}` (nombre réel pour des multiplications).

`\widthof{texte}`, `\heightof{texte}`, `\depthof{texte}`, `\totalheightof{texte}` permettent de récupérer les dimensions de la boîte définie par *texte* :

```
\setlength{\lgrlocal}{\widthof{AA}}
```

assigne la largeur de “AA” à `\lgrlocal`

Les commandes `\maxof{-}{-}` et `\minof{-}{-}` retournent respectivement les maxima et minima des expressions données dans leurs deux arguments (réels).

.....

Le *package* `ifthen`

La comparaison de différents éléments est possible en \TeX .

Le *package* `ifthen` simplifie les expressions conditionnelles.

Il définit la commande `\ifthenelse{-}{-}{-}` où :

1. le premier argument est une condition à tester ;
2. le second argument est le code à exécuter si la condition est vraie ;
3. le troisième argument est le code à exécuter si la condition est fausse.

Les conditions peuvent être construites avec des parenthèses `\(, \)` et les commandes `\and`, `\or` et `\not`. Les atomes élémentaires des conditions sont :

- `\num1<\num2`, `\num1=\num2`, `\num1>\num2`,
- `\isodd{\num}`,
- `\isundefined{\cmd}`,
- `\equal{texte1}{texte2}`,
- `\lengthtest{\dim1<\dim2}`, `\lengthtest{\dim1=\dim2}`,
- `\lengthtest{\dim1>\dim2}`,
- `\boolean{nom}`.

Les commandes `\newboolean{-}` et `\setboolean{-}{-}` permettent de créer des booléens et de leur assigner la valeur *true* ou *false*.

On peut manipuler des boucles avec la commande `\whiledo{test}{code}`.

Le *package* `xifthen` ajoute d'autres tests.

.....

Le *package* `snapshot`

Le *package* `snapshot` répond à une question pratique : quels fichiers fournir ou conserver avec le document source pour que le tout compile correctement ?

La réponse dépend bien sûr de la machine sur laquelle compiler le document.

snapshot s'occupe de collecter dans un fichier `.dep` (dépendances) tous les fichiers appelés lors de la compilation, ainsi que leur version.

Pour l'utiliser, il faut le placer avant tout le reste :

```
\RequirePackage{snapshot}
\documentclass[12pt]{article}
...
```

Après compilation, le fichier `.dep` contient des informations du type

```
*{application}{TeX}      {1990/03/25 v3.x}
*{format} {LaTeX2e}      {2009/09/24 v2.e}
*{class}  {article}      {2007/10/19 v1.4h}
*{file}   {size12.clo}   {2007/10/19 v1.4h}
*{package}{inputenc}     {2008/03/30 v1.1d}
*{file}   {utf8.def}      {2008/04/05 v1.1m}
*{package}{textcomp}     {2005/09/27 v1.99g}
*{file}   {ts1enc.def}    {2001/06/05 v3.0e}
*{package}{amsmath}      {2000/07/18 v2.13}
```

.....

Le moteur `bundledoc` : tirer profit de `snapshot`

On peut utiliser les informations fournies par **snapshot** dans le fichier `.dep` pour en faire automatiquement une archive, grâce au script PERL appelé `bundledoc`.

La commande

```
bundledoc --verbose snapshot.dep
```

construit une archive qui contient tous les fichiers utiles à la compilation, y compris les images et les fichiers de style personnels.

Il est donc possible de réaliser un “instantané” des fichiers utilisés dans un projet pour pouvoir le recompiler plus tard (des années ?).

De nombreuses options sont disponibles.

On peut utiliser un fichier de configurations, on peut inclure des types de fichiers particuliers, on peut exclure des dossiers.

```
bundledoc --exclude=/usr/local/texlive/ ...
```

exclut les fichiers déjà présents dans l'installation **TeXLive**.

➔ très utile pour envoyer les documents sur les serveurs en ligne, comme **arXiv**⁹⁰.

6.2 La gestion de la couleur

Le *package color*





Le *package color* permet de gérer les couleurs dans L^AT_EX.

Comme pour les graphiques, *tex* ne gère pas directement les couleurs : ce sont les drivers qui s'en occupent. Il faut donc préciser quel chemin de production on utilise :

`\usepackage{dvips}{color}` charge les spécificités du driver dvips.

Autres options possibles : dvipdfm, pdftex, xetex...


Une couleur est définie à partir d'un modèle et d'une spécification :


<i>red-green-blue</i>	rgb	3 nombres compris entre 0 et 1 0.5, 0.2, 0.8 → 
<i>cyan-magenta-yellow-black</i>	cmyk	4 nombres compris entre 0 et 1 0.5, 0.2, 0.8, 0.3 → 
Échelle de gris	gray	1 nombre compris entre 0 et 1 0.8 → 
Nom explicite	named	nom d'une couleur prédéfinie magenta → 

La commande

`\definecolor{nom}{modèle}{spécification}`



définit une couleur nommée *nom* dans le modèle choisi.

`\definecolor{light-blue}{rgb}{0.8,0.85,1}` → 

`\definecolor{mygrey}{gray}{0.75}` → 

Utilisation des couleurs

Plusieurs commandes sont définies pour utiliser les couleurs.








- `\color{couleur}` installe la couleur dont le nom est dans l'argument.
Noir `\color{magenta} couleur` et noir → Noir couleur et noir
- `\textcolor{couleur}{texte}` imprime le texte dans la couleur sélectionnée.
Noir `\textcolor{magenta}{couleur}` et noir → Noir couleur et noir
- `\colorbox{couleur}{texte}`, `\fcolorbox{couleur1}{couleur2}{texte}` produisent des boîtes de fond coloré.
`\colorbox{magenta}{test}` → 
`\fcolorbox{blue}{magenta}{test}` → 
- Les longueurs `\fboxrule` et `\fboxsep` gèrent l'épaisseur du trait et de la séparation avec le texte (mêmes paramètres que pour `\framebox`).
- `\pagecolor{couleur}` colore le fond de la page.
- Ces commandes acceptent une variante qui spécifie la couleur par un modèle :
`\color[modèle]{spécification}`
`\textcolor[modèle]{spécification}{texte}`
`\colorbox[modèle]{spécification}{texte}`
`\fcolorbox[modèle]{spécification1}{spécification2}{texte}`
`\pagecolor[modèle]{spécification}`

Le *package xcolor*

Le *package xcolor* étend sur de nombreux points le *package color* en fournissant d'autres modèles et une procédure pour créer des couleurs par mélanges.

Il utilise essentiellement les mêmes options que *color*.






Les nouveaux modèles sont :

<i>cyan-magenta-yellow</i>	cmy	3 nombres compris entre 0 et 1 0.5, 0.2, 0.8 → 
<i>hue-saturation-brightness</i>	hsb	3 nombres compris entre 0 et 1 0.5, 0.2, 0.8 → 
Longueur d'onde en nm	wave	1 nombre compris entre 363 et 814 650.57 → 
<i>Red-Green-Blue</i>	RGB	3 nombres entiers compris entre 0 et 255 100, 150, 200 → 
<i>RRGGBB</i>	HTML	1 nombre hexadécimal entre 000000 et FFFFFFFF 43ADF9 → 
<i>Hue-Saturation-Brightness</i>	HSB	3 nombres entiers entre 0 et 240 10, 100, 200 → 
Niveau de gris	Gray	1 nombre entier entre 0 et 15 10 → 

Il reprend et étend les commandes définies par *color* : `\color`, `\textcolor`, `\colorbox`, `\fcolorbox` et `\pagecolor`.

Les couleurs prédéfinies et les mélanges




Le *package xcolor* définit les couleurs suivantes :


black 	white	red 	green 	blue 
cyan 	magenta 	yellow 	orange 	pink 
brown 	olive 	purple 	teal 	violet 
gray 	darkgray 	lightgray 		

Les options `dvipsnames`, `svgnames` et `x11names` de *xcolor* chargent de nombreuses autres couleurs prédéfinies. Consulter la documentation de *xcolor*⁹¹.

La commande `\definecolor` admet la même syntaxe que pour *color*.

La commande `\colorlet{couleur}{mélange}` définit une couleur à partir de mélanges :

`blue!85!black` → , `blue!65!black` → , `blue!45!black` → 

`-red` →  (couleur complémentaire)

`red!50!green!60!yellow` → , `red!50!green!20!yellow` → 

`red!50!green!20` →  (dernière couleur : white)

Les commandes `\color`, `\textcolor`, `\colorbox`, `\fcolorbox` et `\pagecolor` acceptent directement des couleurs sous forme de mélanges :

`\textcolor{blue!85!black}{test}` → test

Le *package xcolor* fournit beaucoup d'autres fonctionnalités, en particulier la possibilité de transformer une couleur d'un modèle dans un autre.

L'option `gray` convertit toutes les couleurs en niveaux de gris → impression en N & B.

6.3 La gestion des tableaux

Rappels sur les tableaux

L'environnement `tabular` définit les commandes suivantes pour le formatage des colonnes d'un tableau :

- `|` insère une ligne verticale
- `l` alignement à gauche
- `r` alignement à droite
- `c` alignement au centre
- `p{<dim>}` paragraphe de largeur donnée
- `@{code}` supprime l'espace entre colonnes et insère `code`
- `*{num}{opts}` répète `num` fois la déclaration `opts`

Les paramètres suivants déterminent l'aspect du tableau :

- `\tabcolsep` moitié de la largeur de l'espacement entre les colonnes (6pt)
- `\arrayrulewidth` épaisseur des traits du tableau (0.4pt)
- `\doublerulesep` séparation entre les doubles traits (| |) (2pt)
- `\arraystretch` fraction avec laquelle l'espace entre lignes est multiplié (1.0)

Par exemple, `\renewcommand{\arraystretch}{1.5}` écarte les lignes de 50%.

Le package array

Le package **array** définit d'autres commandes de formatage :

- `m{<dim>}` colonne de largeur donnée, verticalement centrée
- `b{<dim>}` colonne de largeur donnée, verticalement en bas
- `>{code}` insère `code` avant les données d'une cellule
- `<{code}` insère `code` après les données d'une cellule
- `!{code}` insère `code` entre deux colonnes

La commande `\newcolumntype{-}{-}` permet de définir des nouveaux types de colonnes en utilisant la syntaxe des déclarations de colonnes.

La dimension `\extrarowheight` s'ajoute à la hauteur des cellules.

```
\setlength{\extrarowheight}{4pt}
\begin{tabular}{|>\large c|>\bfseries l|}
\hline A & B \\
\hline 100 & 50 \\
\hline
\end{tabular}
```

A	B
100	50

```
\newcolumntype{R}{>{\$}r<{\$}}
\begin{tabular}{|R!{=}>\bfseries l|}
\hline a_1 & A \\
\hline b_1 & B \\
\hline
\end{tabular}
```

a_1	A
b_1	B

Le package array : autres exemples

```
\begin{tabular}{p{2em}p{2em}}
\hline a a a a a & b \\
\hline a a a
\end{tabular}
```

a a a	a a a
a a a	b

```
\begin{tabular}{m{2em}m{2em}}
\hline a a a a a & b \\
\hline a a a
\end{tabular}
```

a a a	a a a
a a a	b

```
\begin{tabular}{b{2em}b{2em}}
\hline a a a a a & b \\
\hline a a a
\end{tabular}
```

a a a	b b b b
a a a	b b b

Les commandes `\raggedright` et `\centering` peuvent aussi être utilisées.
⚠ Ces commandes redéfinissent `\`, qu'il faut rétablir aussitôt avec `\arraybackslash`.

Le package tabularx

L'environnement `tabular*` admet un paramètre supplémentaire qui désigne la largeur finale du tableau. Pour atteindre ce but, \LaTeX joue sur l'espace entre les colonnes.

Le package **tabularx** résout ce problème différemment : il définit l'environnement `tabularx` qui admet pour paramètre supplémentaire la largeur finale du tableau.

Pour atteindre cette largeur, un nouveau type de colonne est défini :

X colonne de largeur automatiquement calculée

```
\begin{tabularx}{10em}{|X|X|p{1.1em}|}
\hline a a a a a & b b b b b & b b b b b & c c c c c \\
\hline
\end{tabularx}
```

a a a a a	b b b b b	b b b b b	c c c c c
a a	b b b	b b b	c c

Toutes les colonnes de type X dans un tableau auront la même largeur.

tabularx est compatible avec **array**.

On peut définir de nouveaux types de colonnes avec `\newcolumntype{-}{-}` :

`\newcolumntype{Y}{>\raggedright\arraybackslash}X`

Par définition, X est construit sur `p{<dim>}` où `<dim>` est déterminée par l'algorithme de création du tableau. On peut redéfinir X pour utiliser `m{<dim>}` ou `b{<dim>}` :

`\renewcommand\tabularxcolumn[1]{m{#1}}` redéfinit X pour utiliser `m{<dim>}`.

Couleurs dans les tableaux

- Le package `colortbl` permet de colorer les tableaux.
- `\columncolor{couleur}` colore le fond d'une rangée ;
 - `\rowcolor{couleur}` colore le fond d'une ligne entière ;
 - `\cellcolor{couleur}` colore le fond d'une cellule ;
 - `\arrayrulecolor{couleur}` colore les traits ;
 - `\doublerulesepcolor{couleur}` colore l'espace entre les doubles traits.

```
\arrayrulecolor{green}\doublerulesepcolor{yellow}
\setlength{\arrayrulewidth}{2pt}
\begin{tabular}{|>\columncolor{brown}}r|l|}
\hline \rowcolor{red} AA & BB \\ \hline
CC & DD \\ \hline
EE & \cellcolor{cyan} FF \\ \hline
\end{tabular}
```

AA	BB
CC	DD
EE	FF

On peut utiliser la commande `\newcolumnntype{-}{-}` avec ces commandes. L'option `table` de `xcolor` appelle `colortbl` et définit une commande `\rowcolor[commandes]{ligne}{couleur1}{couleur2}` où les deux couleurs concernent les lignes impaires et paires, *ligne* désigne la première ligne où commencer l'alternance et *commandes* les commandes entre les lignes (`\hline` par exemple).

Autres packages pour créer des tableaux

Les packages `supertabular` (voir aussi son extension `xtab`) et `longtable` permettent de créer des tableaux s'étendant automatiquement sur plusieurs pages.

`hhline` permet de gérer les intersections des lignes horizontales et verticales doubles.

`arydshln` permet d'utiliser des traits pointillés comme séparations de cellules.

`multirow` définit une commande pour regrouper des cellules d'une même colonne.

`booktabs` introduit les commandes `\toprule`, `\midrule` et `\bottomrule` qui gèrent des espacements verticaux moins resserrés que `\hline` :

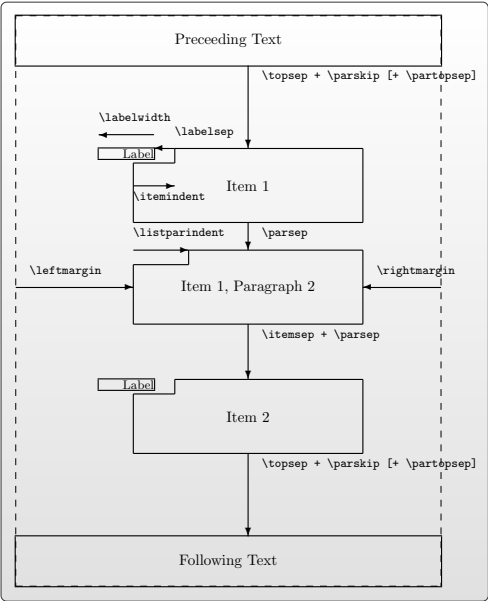
```
\begin{tabular}{@{}l r@{}}
\toprule
\multicolumn{2}{c}{AAA} & 
\multicolumn{1}{c}{BBB} \\ \midrule
CCC & DDD & EE \\
aa & bb & cc \\
dd & ee & ff \\ \bottomrule
\end{tabular}
```

AAA		BBB
CCC	DDD	EEE
aa	bb	cc
dd	ee	ff

Il est utile de consulter la documentation de tous ces packages.

6.4 La gestion des listes

Les listes : la géométrie



- Géométrie des listes
- Les dimensions verticales sont :
- `\topsep`
 - `\partopsep`
 - `\parsep`
 - `\itemsep`
- Les dimensions horizontales sont :
- `\leftmargin`, `\rightmargin`
 - `\labelwidth`
 - `\labelsep`
 - `\itemindent`
 - `\listparindent`
- ➔ pas facile à personnaliser !

Les listes : personnalisation

L'environnement `list` permet de définir des nouveaux types de listes, mais il faut maîtriser à la fois la géométrie décrite auparavant et les labels. Une telle définition peut ressembler à ceci :

```
\newenvironment{Description}
{
\begin{list}{}{\let\makeLabel\Descriptionlabel
\setlength{\labelwidth}{40pt}%
\setlength{\leftmargin}{\labelwidth+\labelsep}}%
\end{list}}
\newcommand*{\Descriptionlabel}[1]{\textsf{#1}\hfil}
```

La première partie définit l'environnement `Description` avec des dimensions spécifiques. La commande `\Descriptionlabel` sert à placer les labels de la liste.

On a déjà vu que le package `pifont` définit les listes `\begin{dinglist}{(num)} ... \end{dinglist}` `\begin{dingautolist}{(num)} ... \end{dingautolist}` où *(num)* est le numéro d'un des symboles de la police ZAPF DINGBATS. La seconde liste est une énumération, elle incrémente *(num)* à chaque `\item`.

`pifont` définit aussi les listes suivantes qui acceptent n'importe quelle famille de police :

```
\begin{Pilist}{fmy}{(num)} ... \end{Pilist}
\begin{Piautolist}{fmy}{(num)} ... \end{Piautolist}
```

.....

Le *package* `enumitem`

Le *package* `enumitem` aide à modifier certains paramètres des 3 listes usuelles `itemize`, `enumerate` et `description`.

```
\begin{enumerate}[label=\emph{\alph*}], parsep=2ex]
  \item ...
  \item ...
\end{enumerate}
```

Il est possible de définir de nouvelles listes :

```
\newlist{maliste}{enumerate}{3}
\setlist{maliste,1}[label=\alph*], font=\sffamily\bfseries}
\setlist{maliste,2}[label=\arabic*], font=\sffamily\bfseries}
```

Les dimensions modifiables sont `topsep`, `partopsep`, `parsep`, `itemsep`, `leftmargin`, `rightmargin`, `listparindent`, `labelwidth`, `labelsep`.

On peut personnaliser la police avec `font=...` et le label avec `label=...`

⚠ Ce *package* n'est pas compatible avec l'option `french` de **`babel`**.

Après l'appel de **`babel`**, il faut utiliser la commande

```
\frenchbsetup{StandardLists=true}
```

pour que `french` ne change pas les listes de L^AT_EX.

.....

Le *package* `paralist`

Le *package* `paralist` permet de modifier facilement les items des listes usuelles à l'aide d'une syntaxe très simple :

```
\begin{enumerate}[\example a)]...
\begin{enumerate}[\A-1]...
\begin{enumerate}[\bfseries {Item} I]...
```

qui produisent :

exemple a), exemple b), exemple c)...

A-1, A-2, A-3...

Item I, Item II, Item III...

D'autres environnements de listes sont définis :

- `inparaenum`, `inparaitem`, `inparadesc` qui produisent des listes dans le même paragraphe ;
- `compactenum`, `compactitem`, `compactdesc` qui produisent les listes usuelles en versions plus compactes ;
- `asparaenum`, `asparaitem`, `asparadesc` qui produisent des listes où chaque item commence un nouveau paragraphe (indentation ordinaire d'un début de paragraphe).

Ce *package* comporte bien d'autres commandes de personnalisation utiles.

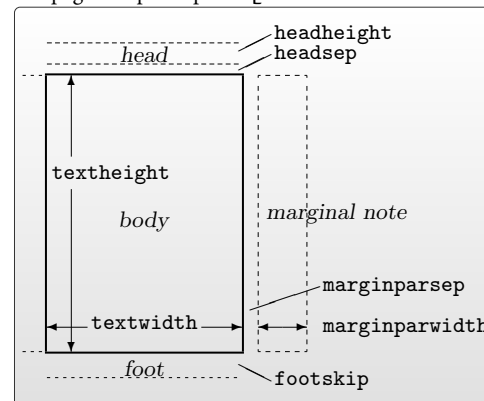
.....

6.5 La mise en page

.....

Rappels sur les éléments d'une page et d'un document

Une page composée par L^AT_EX est constituée de différents éléments :



- l'entête, où figurent souvent le rappel des titres de chapitres et des sections ;
- le pied de page, où figurent souvent le numéro de la page ;
- des notes de bas de page (`\footnote{-}`) ;
- des notes dans la marge (`\marginpar{-}`) ;
- et le corps du texte qui contient à son tour des paragraphes, divisions par chapitres et sections, etc.

.....

Le style des pages

L^AT_EX introduit la commande `\pagestyle{-}` pour installer un style qui va gérer les entêtes et pieds de page.

La commande `\thispagestyle{-}` n'affecte que la page en cours.

Par défaut, 4 styles sont prédéfinis :

`empty` L'entête et le pied de page sont vides.

`plain` L'entête est vide, le pied de page contient le numéro de la page centré.

`headings` L'entête contient les chapitres et sections (ou sous divisions) ainsi que le numéro de la page, le pied de page est vide.

`myheadings` L'entête contient le numéro de la page et des informations fournies par l'utilisateur, le pied de page est vide.

Dans la classe **`book`**, un chapitre commence dans le style `plain`, les autres pages sont dans le style `headings` par défaut.

Les commandes `\markboth{mark1}{mark2}` et `\markright{mark3}` servent à remplir l'entête dans les styles `headings` et `myheadings`.

L'esprit de ces commandes est que `mark1` correspond à une entrée principale, et `mark2` et `mark3` à des entrées secondaires.

Les commandes `\chapter`, `\section...` utilisent ces commandes (ou plutôt des commandes équivalentes) en y insérant le libellé des chapitres, sections...

Les commandes `\leftmark` et `\rightmark`

Il est possible de récupérer des données insérées dans `\markboth{mark1}{mark2}` et `\markright{mark3}` : on dispose des commandes `\leftmark` et `\rightmark`.

Une fois la page terminée, la situation est la suivante :

- `\leftmark` contient la valeur *mark1* du *dernier* `\markboth` rencontré sur la page ➔ entrée principale.
- `\rightmark` contient la valeur *mark2* ou *mark3* du *premier* `\markboth` ou `\markright` rencontré depuis le début de la page, sinon du dernier de la page précédente ➔ entrées secondaires.

En plus clair (?), dans le cas d'un livre : lorsque la page se termine, `\leftmark` récupère le nom du chapitre qui a été créé en dernier, alors que `\rightmark` récupère le nom de la première section créée en début de page (ou de la dernière section créée sur la page précédente).

Le *package* **extramarks** définit des nouvelles commandes pour récupérer les autres entrées des commandes `\markboth` et `\markright` :

- `\firstleftmark` récupère *mark1* du *premier* `\markboth` rencontré sur la page ;
- `\lastrightmark` récupère *mark2* ou *mark3* du *dernier* `\markboth` ou `\markright` rencontré sur la page.

Ces commandes peuvent servir à définir de nouveaux styles de pages.

Le *package* **fancyhdr**

Le *package* **fancyhdr** définit un nouveau style de page appelé *fancy*.

Ce style est composé de 6 parties : les parties gauches, centrées et droites de l'entête et du pied de page, qu'on peut personnaliser à l'aide des 2 commandes :

`\fancyhead[-]{-}` et `\fancyfoot[-]{-}`.

L'argument optionnel utilise les lettres L, C, R pour désigner l'une des trois zones, et O, E pour désigner les pages impaires ou paires.

On peut définir l'aspect des lignes sous l'entête et au dessus du pied par les commandes `\headrule` et `\footrule`, et leur épaisseur par les commandes `\headrulewidth` et `\footrulewidth`.

```
\pagestyle{fancy} \fancyhead{} \fancyfoot{}
\fancyhead[LO]{\slshape\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\fancyfoot[LO,RE]{\thepage}\fancyfoot[RO,LE]{\today}
\renewcommand{\headrule}{\dotfill}
\renewcommand{\footrulewidth}{0pt}
```

La commande `\fancypagestyle{-}{-}` permet de redéfinir un style déjà défini.

Si on n'utilise pas l'option *twoside*, on peut utiliser les 6 commandes `\lhead{-}`, `\chead{-}`, `\rhead{-}`, `\lfoot{-}`, `\cfoot{-}` et `\rfoot{-}`.

Lire la documentation pour plus d'informations.

Le *package* **titlesec** : le style des pages

Le *package* **titlesec** permet de (re)définir des styles de page.

La commande `\newpagestyle{nom}[options globales]{code}` et son équivalent `\renewpagestyle` (re)définissent un style nommé *nom* selon le code donné par *code*. Les *options globales* s'appliquent aux entêtes et aux pieds. Le code *code* repose sur deux commandes `\sethead[-][-][-]{-}{-}{-}` et `\setfoot[-][-][-]{-}{-}{-}`.

Ces lots de trois cases correspondent aux parties gauches, centrées et droites des entêtes et des pieds. Les parties optionnelles concernent les pages paires (voir les arguments comme sur un livre ouvert sur deux pages).

De nombreuses commandes permettent d'insérer dans ces arguments des données diverses :

- `\headrule`, `\footrule`, et variantes... pour des filets ;
- `\chaptertitle`, `\sectiontitle`... pour les titres de chapitres...
- `\ifthechapter{-}{-}`, `\ifthesection{-}{-}`...

Des options du *package* permettent de choisir le type des *marks* :

`outermarks`, `innermarks`, `topmarks`, `botmarks`.

On peut aussi directement désigner ces *marks* grâce aux commandes :

`\bottitlemarks`, `\toptitlemarks`, `\firsttitlemarks`, `\nexttoptitlemark`, `\outertitlemarks`, `\innertitlemarks`.

Le *package* **titlesec** : exemples de styles

Rédefinition du style *plain* pour insérer la date et mettre le numéro de page à l'extérieur de la page, le tout en gras et dans la famille sans-sérif :

```
\renewpagestyle{plain}
[\normalfont\sffamily\bfseries\mathversion{bold}]
{\setfoot[\today][[]]{\today}
\sethead[\usepage][[]]{\usepage}}
```

Définition d'un nouveau style :

```
\newpagestyle{main}
[\normalfont\sffamily\bfseries\mathversion{bold}]
{\headrule
\setfoot[\today][[]]{\today}
\sethead[\usepage][[]]{\ifthechapter%
{\chaptertitlename\thechapter - \chaptertitle}
{\chaptertitle}}
{\ifthesection{\thesection - \sectiontitle}
{\chaptertitle}}%
}{\usepage}}
```


.....

Le *package* `titlesec` : les commandes de section

La finalité première du *package* `titlesec` est de personnaliser l'apparence des commandes de type sections.

Deux commandes sont définies pour ça :

- `\titleformat{-}[-]{-}{-}{-}{-}[-]` pour décrire comment rendre un commande du type `\chapter`, `\section`...
- `\titlespacing*{-}{-}{-}{-}{-}[-]` pour définir les espacements avant et après les commandes `\chapter`, `\section`...

Les arguments de `\titleformat` désignent des points précis comme : la forme générale du label (nouveau paragraphe, en ligne...), le formatage général d'un point de vue typographique, le formatage du label seul, le formatage du titre seul, les espaces horizontaux entre le label et le titre...

Les arguments de `\titlespacing*` concernent les espacements horizontaux à gauche et à droite, et les espacements verticaux avant et après.

Des commandes pour la justification et pour placer des filets sont fournies.

Des commandes plus légères sont aussi fournies pour personnaliser plus simplement les commandes `\chapter`, `\section`...

➔ Consulter la documentation pour des explications plus fournies et où de nombreux exemples sont donnés.

À noter que le *package* compagnon `titletoc` permet de personnaliser la table des matières dans le même esprit.

.....

Le *package* `titlesec` : exemples

Redéfinition du style des chapitres :

```
\titleformat{\chapter}[display]
  {\raggedright\normalfont\sffamily\LARGE\bfseries%
   \mathversion{bold}}
  {\chaptertitlename\ \thechapter}{0pt}{\huge}

\titlespacing*{\chapter}{0pt}{20pt}{4ex plus 1ex minus 1ex}
\assignpagestyle{\chapter}{empty}
\newcommand{\chapterbreak}{\cleardoublepage}
```

Redéfinition du style des sections :

```
\titleformat{\section}
  {\normalfont\sffamily\large\bfseries\mathversion{bold}}
  {\thesection}{1em}{}

\titlespacing*{\section}{0pt}{3ex plus 2.5ex minus .8ex}
{2ex plus .5ex}
```

6.6 Autres éléments du document

.....

La page de garde

Dans la classe `article`, la commande `\maketitle` produit une page de garde avec des données renseignées auparavant par l'auteur à l'aide des commandes `\title{-}`, `\author{-}`, `\date{-}` et `\thanks{-}`.

On peut créer soi-même une mise en page convenable pour réaliser une page de garde.

L'environnement `titlepage` de L^AT_EX crée une page de titre de style vide, et la page suivante commence avec le numéro 1.

À l'intérieur de cet environnement on peut faire ce qu'on veut.

Le document **Some Examples of Title Pages**⁹² fournit un grand nombre d'exemples de pages de gardes, avec le code pour les réaliser.

Le *package* `titling` permet de personnaliser le rendu de la commande `\maketitle`.

Il définit des commandes `\pretitle{-}`, `\posttitle{-}`, `\preauthor{-}`, `\postauthor{-}`, `\predate{-}`, `\postdate{-}` qui insèrent du code avant et après les éléments spécifiés : `{\pretitle... \title... \posttitle}`.

```
\pretitle{\begin{flushright}\LARGE\sffamily\bfseries}
\posttitle{\par\end{flushright}\vskip 0.5em}
```

Il est même possible d'insérer du code supplémentaire entre les éléments du titre par des commandes du type `\maketitlehooka{-}`... `\maketitlehookd{-}`.

Il définit l'environnement `titlingpage` qui compose la page de garde sur une page à part.

.....

Le *package* `fancybox`

Par défaut, L^AT_EX définit une commande pour encadrer du texte :

`\fbox{-}` ➔ **Primum movens** (Voir aussi `\framebox` avec ses options.)

Le *package* `fancybox` définit une série de commandes et d'environnements destinés à encadrer des éléments de la page : du texte, des listes, des flottants, des formules de mathématique, et même toute la page !

Les boîtes définies grâce à `fancybox` sont :

`\shadowbox{-}` ➔ **Primum movens**

`\doublebox{-}` ➔ **Primum movens**

`\ovalbox{-}` ➔ **Primum movens**

`\Ovalbox{-}` ➔ **Primum movens**

De nombreux paramètres permettent de gérer l'aspect de ces encadrements.

Remarque technique : Pour les amateurs de programmation, `fancybox` définit aussi des commandes pour la gestion des environnements de type verbatim...

On a déjà vu que le *package* `empheq` pouvait utiliser les boîtes de `fancybox` en mode mathématique.

Packages en vrac

Le package **landscape** définit l'environnement `landscape` qui bascule son contenu en format paysage, tandis que l'entête et le pied restent dans le format portrait. Le contenu peut parcourir plusieurs pages.

Le package **soul** définit des commandes pour modifier l'espacement entre les lettres, passer en petites capitales, souligner, barrer, surligner et mettre en valeur du texte. Il est possible de personnaliser le comportement des commandes définies.

Le package **lastpage** permet d'accéder au numéro de la dernière page.

Le package **numprint** propose de formater les grands nombres, avec notation scientifique, et gestion des unités physiques. Il permet aussi d'imprimer des longueurs et des compteurs.

Le package **relsize** définit des commandes pour gérer la taille du texte relativement à la taille de l'environnement : `\relsize{-}`, `\smaller[-]`, `\larger[-]`...

Le package **setspace** définit des commandes pour gérer l'espacement entre les lignes.

Les packages **showlabels** et **showkeys** permettent d'afficher les labels attribués par l'auteur. Utile dans la phase préparatoire d'un document. Nombreuses options disponibles.

Le package **listings** permet d'imprimer des codes informatiques, avec coloration syntaxique, numérotation des lignes, encadrement... Il reconnaît de nombreux langages informatiques.

Cours 7 – L^AT_EX et les graphiques

« Ce dessin m'a pris cinq minutes,
mais j'ai mis soixante ans pour y arriver. »
Pierre-Auguste Renoir

7.1 Généralités sur les graphiques	123
Où l'on s'initie à regarder la nature intime d'une image et la structure informatique d'un graphique...	
7.2 Faire des dessins à la souris	124
Où l'on choisit d'abord de voir s'il nous est possible de dompter la souris pour produire de beaux dessins à dessein...	
7.3 Illustrer les mathématiques	129
Où l'on conçoit que les données numériques sont mieux présentées avec des outils adéquats, que les courbes analytiques méritent des logiciels dédiés et que les constructions géométriques usuelles peuvent s'informatiser...	
7.4 Programmer des graphiques	134
Où l'on découvre qu'il est possible de programmer un dessin comme on a appris auparavant à programmer du texte...	
7.5 Insertion de graphiques externes	147
Où l'on apprend l'art élémentaire du collage d'images au cœur de nos textes...	
7.6 Insertion d'une figure avec légende	151
Où l'on découvre quelques commandes pour disposer au mieux nos images légendées dans le texte...	

.....

Documentation à consulter

Documentation générale sur les graphiques et L^AT_EX :

- **The L^AT_EX Graphics Companion** ⁹³, M. Goossens, F. Mittelbach, S. Rahtz, D. Roegel, H. Voss, Addison-Wesley 2007.
- **Graphics in L^AT_EX** ⁹⁴, un article qui recense les *packages* pour produire des graphiques au sein de L^AT_EX.

Documentation pour le logiciel Asymptote :

- **Le site officiel d'Asymptote** ⁶³ donne de nombreux exemples.
- **Asymptote: the Vector Graphics Language** ⁹⁵, la documentation officielle.
- **Asymptote - Galeries d'exemples** ⁹⁶, un site qui fournit des exemples simplifiés illustrant de nombreux points du langage.

Documentation pour le *package* TikZ :

- **Le site officiel des *packages* pgf et tikz** ⁹⁷.
- **TikZ & PGF** ⁹⁸, le manuel officiel.
- **TikZ and PGF examples** ⁹⁹, un site qui collecte de nombreux exemples, une bonne façon de démarrer en faisant du copier-coller...
- **Graphics with TikZ** ¹⁰⁰, un article court qui explique l'essentiel de TikZ.
- **TikZ pour l' impatient** ¹⁰¹, un excellent livre en PDF très simple et très pédagogique.

Les autres logiciels qui seront évoqués ont leur propre documentation, qui sort du stricte cadre d'une installation L^AT_EX.

7.1 Généralités sur les graphiques

.....

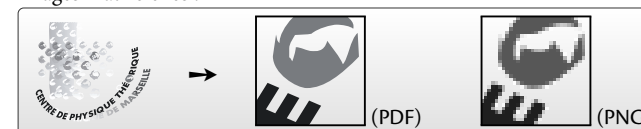
Différents formats d'images

Le monde des images se divise en deux catégories.

Les images matricielles (BITMAP) ¹⁰² Une image matricielle est une image numérique composée d'un tableau de pixels (points de couleurs) et de données globales (taille, résolution...).
Exemples de formats : JPG, PNG, TIFF, BMP...

Les images vectorielles ¹⁰³ Une image vectorielle est une image numérique composée d'objets géométriques individuels (segments de droite, polygones, arcs de cercle, etc.) définis chacun par divers attributs de forme, de position, de couleur, etc.
Exemples de formats : EPS, PDF, SVG...

L'agrandissement des images vectorielles est possible à n'importe quelle échelle, pas celui des images matricielles :



Produire des images BITMAP est relativement facile : photographie numérique, nombreux logiciels de dessin, *scan* d'une figure...

Produire des images vectorielles requiert des logiciels dédiés.

Les logiciels de création de dessin mentionnés dans ce qui suit seront de ce type.

.....

Particularités de certains formats d'images

Les formats d'images PNG, TIFF, PDF et SVG gèrent des "couleurs transparentes".

Lorsqu'on superpose l'image sur un fond coloré (ou une autre image), celle du dessous peut se voir à travers celle du dessus.

```
{\color{blue!35}\rule{1.5cm}{1.5cm}}%
\hspace{-1.5cm}%
\includegraphics[width=1.5cm]{CPT.pdf}
```



La transparence peut être totale ou partielle (on parle d'opacité).

JPG et EPS ne gèrent pas la transparence.

Tous les formats d'images que L^AT_EX utilise doivent impérativement comporter des informations sur la dimension de l'image : L^AT_EX réserve un espace de la taille de l'image (après d'éventuelles transformations). On parle de "BoundingBox".

L'utilitaire `ebb` permet d'afficher la "BoundingBox" des fichiers .pdf, .png et .jpg :

```
ebb -0 image.png
```

L'utilitaire `pdfcrop` découpe un fichier .pdf pour qu'il ait la taille minimale du dessin :

```
pdfcrop image.pdf image-reduite.pdf
```

```
pdfcrop -margins 10 image.pdf image-reduite.pdf
```

→ permet d'ajouter des marges supplémentaires.

7.2 Faire des dessins à la souris

Principes généraux des logiciels de dessins vectoriels

Les logiciels de dessins vectoriels se ressemblent tous dans leurs principes de fonctionnement.

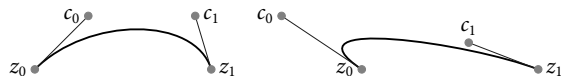
- Une zone de travail aux dimensions personnalisables.
- Possibilité d'agrandir la vue pour travailler plus finement.
- Un ensemble d'objets graphiques pré-définis : cercles, rectangles, arcs de cercle, ovales, étoiles, croix, spirales...
- Dessins de segments et de courbes passant par des points libres.
 - ➔ lignes brisées, régulières, ouvertes, fermées...
- Souvent construites à partir de COURBES DE BÉZIER¹⁰⁴.
- Possibilité de remplir des zones fermées par des couleurs, des dégradés de couleurs...
- Personnalisation des lignes : couleur, épaisseur, pointillés, tiretés, flèches...
- Positionnement d'objets sur la zone de travail : positionnement horizontal, rotation, échelle, positionnement en profondeur (devant ou derrière d'autres objets)...
- Couches de dessins indépendantes qui peuvent se superposer.
- Groupement de plusieurs objets pour les manipuler ensemble.
- Outils de texte plus ou moins riches.
- Copier-coller des différents objets créés.
- Import de fichiers d'images et de dessins, export dans différents formats.

Intermède : les courbes de Bézier

Les courbes de Bézier sont des courbes polynomiales paramétriques qui permettent de dessiner une courbe lisse à partir de la donnée de certains points du plan.

La courbe de Bézier la plus utilisée est une courbe du plan complexe paramétrée par un polynôme de degré 3 :

$$t \mapsto (1-t)^3 z_0 + 3t(1-t)^2 c_0 + 3t^2(1-t)c_1 + t^3 z_1, \quad t \in [0, 1]$$



Les points z_0 et z_1 sont appelés les points initiaux et finaux (ou points extrêmes, nœuds), les points c_0 et c_1 sont appelés les points de contrôle (ou poignées).

De nombreuses variantes de ce paramétrage existent : certains donnent la "tension" entre les points initiaux et finaux, d'autres les directions tangentielles en les points extrêmes, d'autres la "courbure de la courbe" aux points extrêmes.

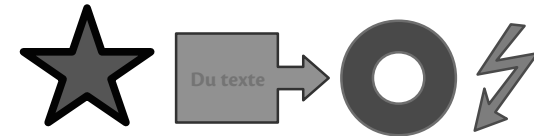
Il existe aussi des courbes de Bézier quadratique de la forme $(1-t)^2 z_0 + 2t(1-t)c + t^2 z_1$, ainsi que des courbes de Bézier linéaires (segment de droite).

Le module de dessin de OpenOffice

OpenOffice¹⁰⁵ est un logiciel libre qui se présente comme une suite bureautique, avec de nombreux modules : traitement de texte, tableur, présentation et surtout dessin.

Les principaux avantages du module Draw¹⁰⁶ de OpenOffice :

- Nombreux objets géométriques prédéfinis.
- Grande simplicité de manipulation des objets graphiques, grande souplesse.
- Format de fichier de sauvegarde ouvert, assuré d'une certaine pérennité, et format d'exportation en PDF pour l'insertion dans L^AT_EX.
- Fonctionne sur tous les systèmes d'exploitations courants.



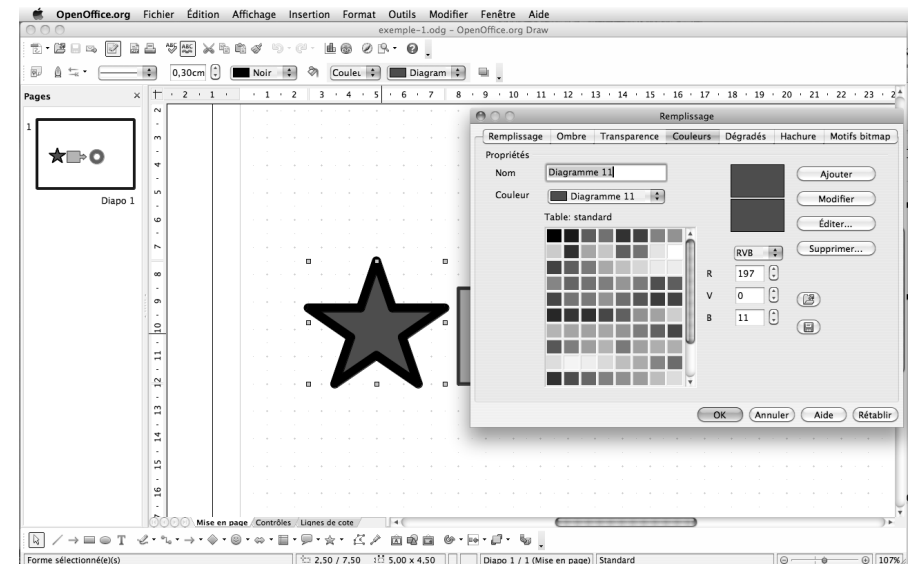
⚠ Suite à un bug qui résiste aux protestations, lorsqu'on exporte un dessin en PDF, il n'a pas la bonne taille (il conserve la taille de la page entière).

➔ on peut lui appliquer le script pdfcrop :

```
pdfcrop image.pdf image-reduite.pdf
```

Faire un dessin avec OpenOffice

L'environnement de travail de OpenOffice :

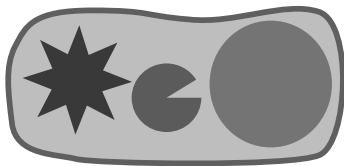


Le logiciel Inkscape

Inkscape¹⁰⁷ est un logiciel libre d'édition de graphismes vectoriels, doté de capacités similaires à des logiciels professionnels du type Illustrator¹⁰⁸.

Les principaux avantages de Inkscape :

- Son format de fichier natif est le SVG (*Scalable Vector Graphics*) qui est un des standards du WEB
→ assuré d'une pérennité.
- Nombreux objets géométriques prédéfinis.
- Outil très professionnel, grande puissance.
- Format d'exportation en PDF pour l'insertion dans L^AT_EX.
- Formats d'importation très divers : POSTSCRIPT, EPS, PDF, JPG, PNG, TIFF...
- Fonctionne sur tous les systèmes d'exploitations courants.
- on peut importer un fichier PDF pour le retravailler.



Inkscape et L^AT_EX

Inkscape est intéressant car il est conçu pour exporter en L^AT_EX.

La procédure à suivre est la suivante :

- Menu "Fichier" > "Enregistrer sous..."
- Choisir "Portable Document Format (PDF)" et cliquer sur "Sauver"
- Cocher les cases "PDF+LaTeX..." et "Exporter le dessin".

Le résultat produit deux fichiers :

un fichier image image.pdf et un fichier texte image.pdf_tex.

Dans un document L^AT_EX, on insère alors l'image par le code suivant :

```
\usepackage{graphicx}
\usepackage{xcolor}
...
\def\svgwidth{5cm}
\input{image.pdf_tex}
```

Le fichier image.pdf_tex s'occupe d'insérer l'image, et en plus il lui superpose les libellés de texte qu'on a placé sur le dessin dans Inkscape.

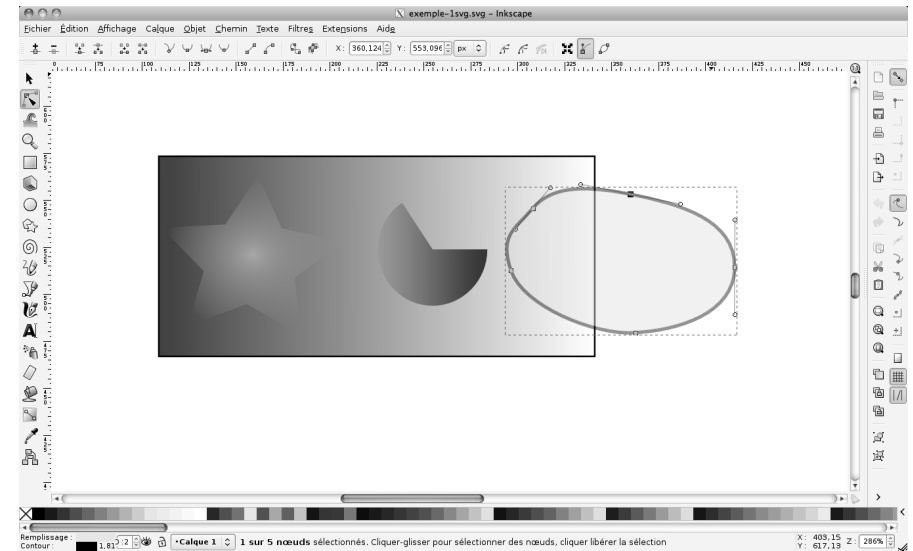
Ces libellés de texte peuvent être du code L^AT_EX quelconque, qui sera compilé dans le document L^AT_EX où l'image est insérée par cette procédure.

On peut éditer le fichier image.pdf_tex pour modifier des libellés de texte.

\svgwidth sert à donner la taille finale de l'image en largeur.

Faire un dessin avec Inkscape

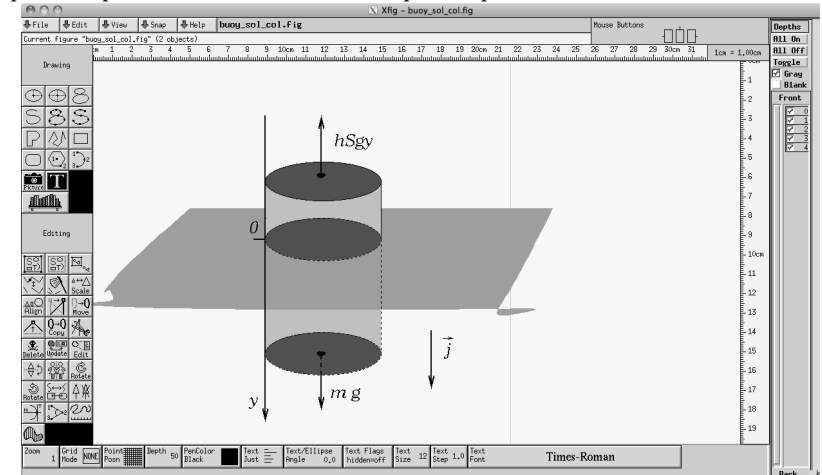
L'environnement de travail de Inkscape :



Le logiciel XFig

XFig¹⁰⁹ est un logiciel de dessin vectoriel compatible sur bien des points avec L^AT_EX.

On peut lui reprocher son interface vieillotte et peu compatible avec les canons actuels.



Le logiciel fig2vect¹¹⁰ permet de convertir des fichiers générés par XFig en des fichiers SVG éditables par Inkscape.

.....
Autres logiciels de dessin à la souris

Il existe de nombreux logiciels de dessins vectoriels, soit gratuits, soit payants.

jfig ¹¹¹ Ce logiciel est une version JAVA de XFig. Il en reprend l'interface, le format de fichier, et les formats d'exportation.
Il est compatible avec L^AT_EX de la même façon que XFig.

Ipe ¹¹² C'est un logiciel libre destiné à la création de dessins vectoriels à insérer dans L^AT_EX.
Les objets de texte peuvent contenir des commandes L^AT_EX.

Adobe Illustrator ¹⁰⁸ (payant) C'est le logiciel de référence en matière de dessins vectoriels. Il est intégré à la suite **Adobe Creative Suite**, qui contient **Adobe Photoshop** ¹¹³ (dessins BITMAP) et **Adobe InDesign** ⁴⁷ (mise en page).

CorelDRAW ¹¹⁴ (payant) C'est le concurrent direct du logiciel précédent. Il est accompagné d'un logiciel de retouche photographique.

sk1 illustration program ¹¹⁵ Ce logiciel libre est encore en développement.
Son but est d'obtenir un logiciel multiplateforme comparable à **Adobe Illustrator** et **CorelDRAW**.

D'autres logiciels plus confidentiels et plus spécialisés sont disponibles.
On trouvera sur la page **Comparison of vector graphics editors** ¹¹⁶ une liste commentée de tels logiciels.

On peut désormais utiliser des logiciels en ligne (à l'intérieur d'un navigateur Internet) pour créer des dessins vectoriels, comme par exemple sur le site **Aviary** ¹¹⁷.

7.3 Illustrer les mathématiques

.....
Différents types de logiciels

Différents problèmes mathématiques peuvent s'illustrer selon diverses méthodes, donc selon divers types de logiciels :

Les logiciels à vocations mathématiques : Il s'agit de logiciels utilisés pour manipuler des expressions mathématiques (analyse, algèbre, géométrie...), pour générer et traiter des données numériques (codes de simulations, traitements statistiques, visualisation dynamiques...) :

Mathematica ¹¹⁸, **Maple** ¹¹⁹, **Matlab** ¹²⁰, **Scilab** ¹²¹, **Maxima** ¹²², **Octave** ¹²³, **Magma** ¹²⁴, **Sage** ¹²⁵, **GNUPlot** ¹²⁶...

Les logiciels de type tableurs : Logiciels de manipulation de séries de données, avec diverses possibilités de programmation :

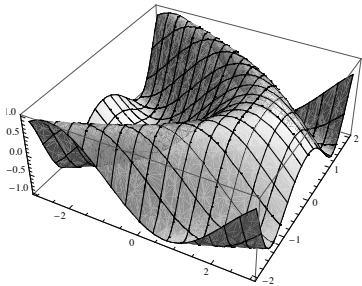
Excel ¹²⁷, **Calc** ¹²⁸ de **OpenOffice**...

Les logiciels de "géométrie dynamique" : Logiciels dont le but est de réaliser des constructions géométriques en se basant sur des relations mathématiques précises :

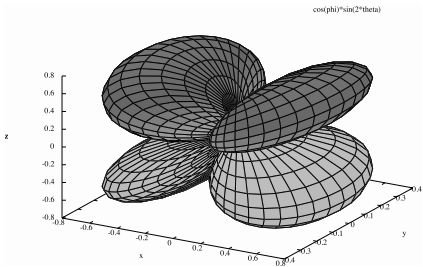
GeoGebra ¹²⁹, **CaRMetal** ¹³⁰...

.....
Les logiciels polyvalents

Mathematica ¹¹⁸, **Maple** ¹¹⁹, **Maxima** ¹²², **Octave** ¹²³... sont des logiciels très polyvalents : calculs formels, manipulation et représentation d'expressions analytiques, traitement de données... Ces logiciels sont prévus pour représenter graphiquement les données et les structures qu'ils manipulent.



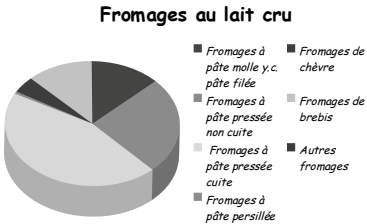
Mathematica



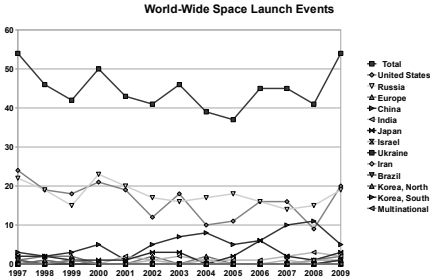
Maxima

Les tableurs

Les logiciels de type tableurs permettent non seulement de traiter des grandes séries de données, mais aussi de représenter graphiquement ces données sans avoir à apprendre un langage spécifique et technique.
→ rapidité de mise en œuvre, personnalisations possibles.



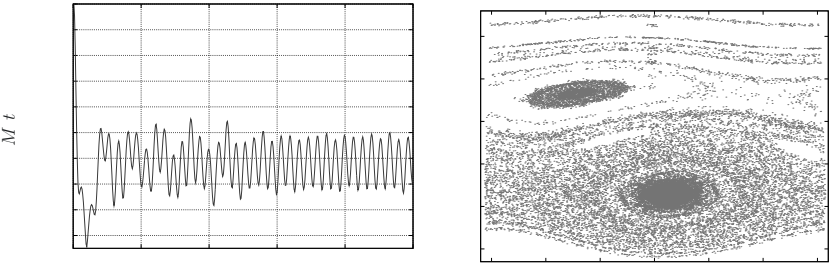
Calc de OpenOffice



Calc de OpenOffice

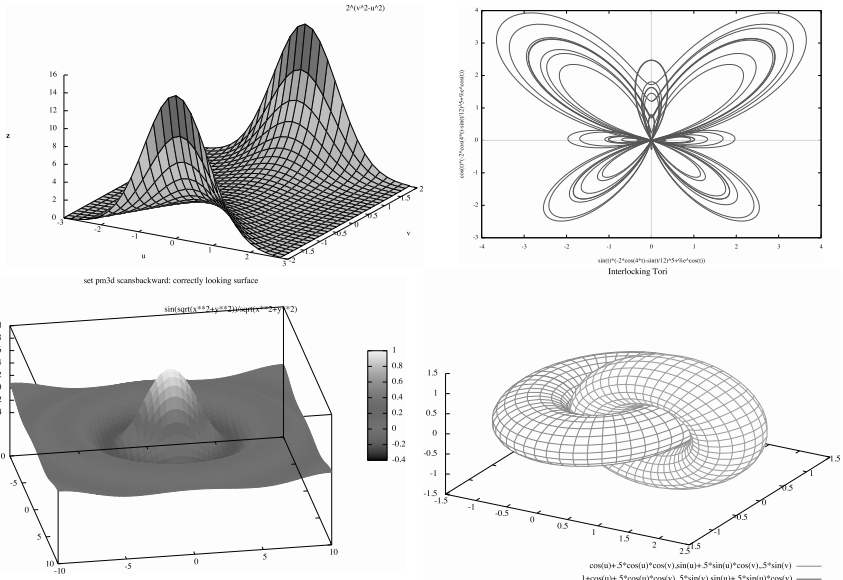
Représentations de données avec GNUPlot

Le logiciel GNUPlot¹²⁶ est très utilisé pour tracer des graphes, soit à partir de fonctions mathématiques, soit à partir de données numériques.
Il est utilisé dans Maxima et Octave pour la génération des graphes.
Il exporte dans de nombreux formats : EPS, PDF, PNG, SVG...
Il peut séparer l'image et les libellés pour que L^AT_EX s'occupe des libellés à la compilation.



(Exemples fournis par Xavier Leoncini)

Représentations analytiques avec GNUPlot



On consultera la documentation pour plus de renseignements.

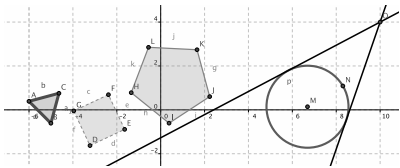
Les logiciels de géométrie dynamique

À l'origine, les logiciels de *géométrie dynamique* se donnent pour but de permettre de réaliser des constructions géométriques du type "à la règle et au compas".
En réalité ils font beaucoup plus : tracé de courbes analytiques, insertion de texte et d'images, programmation de dessins, réalisation de démonstrations dynamiques...
Entités élémentaires : points, droites, segments, demi-droites, cercles, arcs de cercles, polygones réguliers ou non, vecteurs, texte, images...
Opérations géométriques : placer des points au milieu de deux autres ou à l'intersection de deux droites, tracer des parallèles, des perpendiculaires, des médiatrices et des bissectrices, cercles passant par certains points...
Informations et tests : mesures de longueurs et d'angles, coordonnées des points, appartenance d'un point à une droite ou à un cercle, alignement de points, position relatives de deux droites (parallèles ou perpendiculaires)...
Fonctionnalités diverses : possibilité de bouger les objets initiaux (on voit la construction s'actualiser en direct), conservation de l'historique, exportations diverses (internet, images...)...
Utilisation : géométrie plane, géométrie dans l'espace, physique (mécaniques, électricité, astronomie, électromagnétisme)...

⚠ Ce ne sont pas des logiciels de dessin à la souris comme ceux évoqués auparavant : ils ne permettent pas par exemple de tracer des courbes de Bézier de façon native.

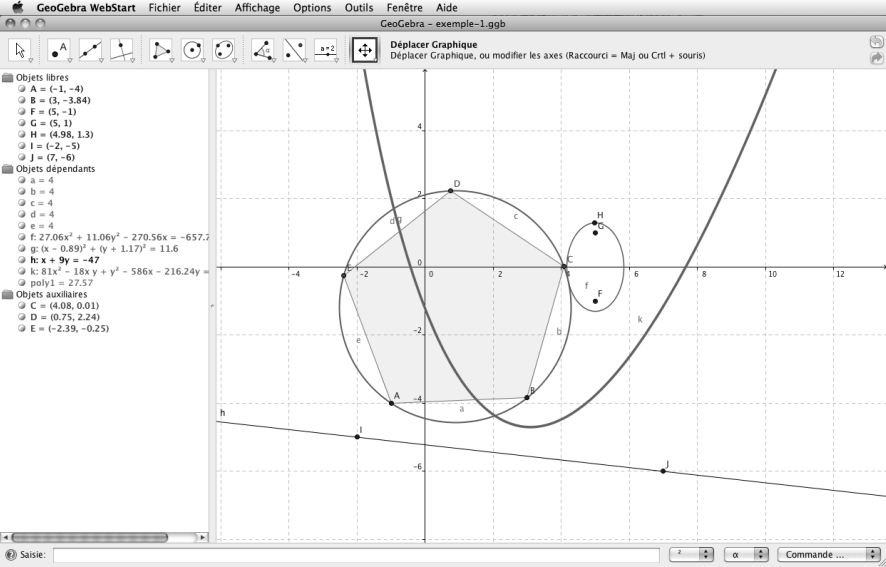
Les logiciels GeoGebra et CaRMetal

- GeoGebra ¹²⁹ et CaRMetal ¹³⁰ sont des logiciels libres de géométrie dynamique.
- Ils sont écrits en JAVA ➔ disponibles sur toutes les plateformes.
 - GeoGebra a des fonctions de tableur.
 - Il est possible de programmer des dessins en JAVASCRIPT, de créer des outils auxiliaires, de créer des interfaces de contrôles.
 - GeoGebra peut exporter en PNG, EPS et dans le format TikZ, ce qui le rend compatible avec L^AT_EX.
 - CaRMetal peut exporter en PNG, EPS et SVG.
 - Ces logiciels sont très utilisés dans le milieu enseignant du secondaire, et aussi au Baccalauréat.



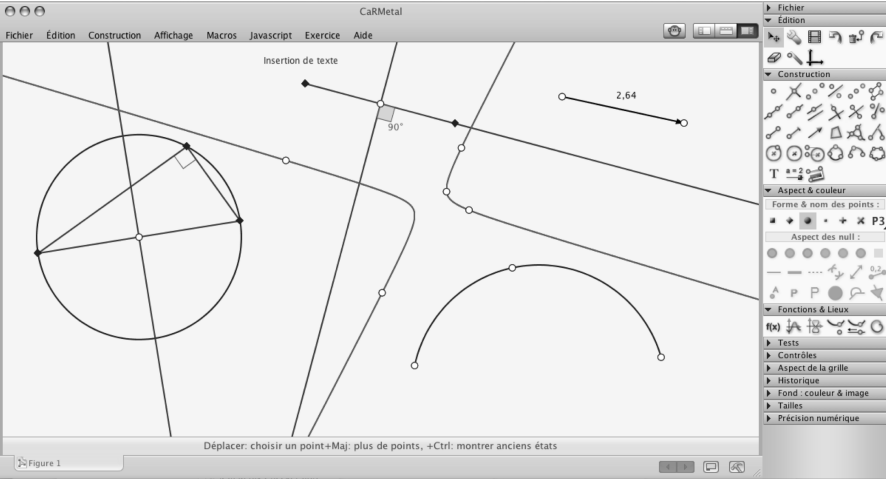
Faire un dessin avec GeoGebra

L'environnement de dessin de GeoGebra :



Faire un dessin avec CaRMetal

L'environnement de dessin de CaRMetal :



7.4 Programmer des graphiques

Les langages de description de dessins vectoriels

Les langages de description de dessins vectoriels peuvent être différents dans leur implémentation, mais ils utilisent les mêmes idées de base :

- Un fichier texte dans lequel le code de description du dessin est écrit.
- Un compilateur qui produit le dessin.
- Des variables qui représentent des longueurs, des épaisseurs de trait, des couleurs, des points dans le plan, voire des points dans l'espace, des chemins...
- Des commandes de dessin élémentaires : dessiner un contour, remplir un contour...
- Des commandes pour réaliser des formes standard : lignes droites, rectangles, cercles, ovales, arc de cercle, étoiles, croix...
- Définir des chemins basés sur les lignes droites et des courbes de Bézier quadratiques ou cubiques.
- Placer du texte sur le dessin, avec des choix typographiques divers, éventuellement le long de chemins compliqués...
- Des transformations géométriques élémentaires : translation, homothétie, réflexion, matrice linéaire...
- La possibilité de sauver un morceau de dessin pour l'utiliser à plusieurs reprises.
- Des commandes de programmation : des boucles, des conditions, des nombres aléatoires, des commentaires...
- Des modules externes qui ajoutent de nombreuses fonctionnalités.

Le logiciel Asymptote

Asymptote⁶³ est issu d'une histoire liée à celle de T_EX :

- En même temps que T_EX, D. E. Knuth crée METAFONT, un moteur et un langage destinés à créer des fontes pour T_EX.
- En 1994, John D. Hobby adapte le langage METAFONT et son moteur pour produire des fichiers EPS en sortie : c'est METAPOST¹³¹.
- Asymptote a été créé en 2002 pour dépasser les limitations de METAPOST : calculs en haute précision, export dans différents formats d'images (EPS, PDF, SVG...).

Principaux avantages de Asymptote :

- Intégration très forte avec L^AT_EX pour placer des libellés dans les dessins, mais aussi pour créer les images (dans un document L^AT_EX).
- Langage de description de dessins très complet avec possibilité de l'enrichir avec des modules : formes usuelles, couleurs, transparences, dégradés, textures, boucles, conditions...
- Possibilité de résoudre des problèmes d'intersections de chemins, de trouver la valeur du paramètre d'une courbe en un point donné, de placer du texte sur un chemin courbe, de superposer des couches...
- Modules très divers : dessins en 3 dimensions, géométrie plane de type "collège" (perpendiculaires, triangles...), diagrammes de Feynman, statistiques, nœuds...
- Installé sur les dernières moutures de TeXLive ➡ accessible à tous par défaut.

Le logiciel Asymptote : utilisation

Il y a trois façons simples d'utiliser Asymptote :

1. Créer un fichier .asy dans lequel figure le code de l'image à composer.

`asy -vv -f pdf image.asy` produit le fichier image image.pdf.
asy est le moteur de compilation de Asymptote.

2. Créer un document L^AT_EX, nommé fichier.tex, avec les commandes

```
\usepackage{asymptote}
...
\begin{asy}
  code de l'image
\end{asy}
```

`pdflatex fichier.tex` ➡ produit des fichiers fichier-n.asy (n numéro),

`asy -vv fichier-*.asy` ➡ produit les images fichier-n.pdf

`pdflatex fichier.tex` ➡ insère les images dans fichier.pdf

3. Créer un document L^AT_EX comme ci-dessus avec l'option

```
\usepackage[inline]{asymptote}
```

et procéder de la même façon. Cette fois les libellés de texte placés dans le code de l'image seront composés par pdf_latex lors de la seconde compilation : ils ne sont pas dans les fichiers images fichier-n.pdf.

Le logiciel Asymptote : exemple

```
import settings;
outformat="pdf";
unitsize(1cm);

pair z=(1,6);
dot(Label("$A$",red),z,(2,2),red+10bp);
draw((1,5)--(1.5,5.5)--(2,5)--(4,6),blue+4bp);
draw((0,2)..(2,4)..(3,3)..(5,3),blue+2bp);

draw((0,0)--(2,2),Arrow);
draw((1,0)--(3,2),red,Arrow(5bp));
draw((2,0)--(4,2),blue,Arrow(SimpleHead,10bp));

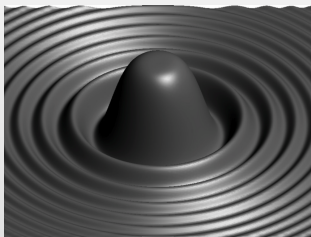
path p=circle((2,-2),1.5);
draw(p,blue+2bp);
draw(shift(1,0)*p,red+1bp);

path q=(0,-4)--(4,-4)--(2,-5)--cycle;
filldraw(q,lightgreen,2bp+.8blue);
```



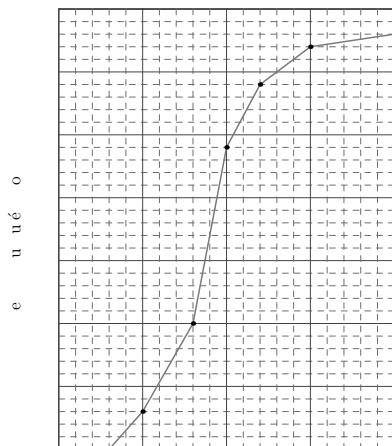
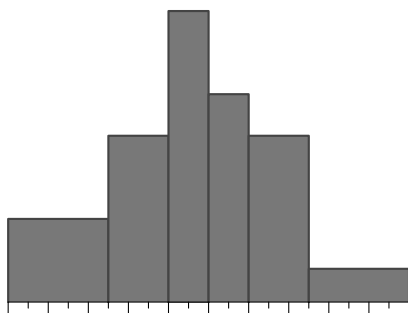
Courbes analytiques avec Asymptote

```
import graph3;
size(7.5cm,0);
currentprojection=orthographic(2,0,1.5);
real f(pair z) {
    real r=2pi*(z.x^2+z.y^2);
    if (r!=0) return sin(r)/r;
    else return 1;}
draw(surface(f,(-2.5,-2),(2.5,2),100,Spline),lightblue);
```



Représentations de données avec Asymptote

Le module stats de Asymptote permet de gérer des données statistiques.



(Exemples extraits de Asymptote - Galeries d'exemples⁷⁹⁶)

Le package TikZ

Le package **tikz** repose sur le package **pgf**, *Portable Graphic Format*.

Origine du nom : TikZ = “TikZ ist *kein* Zeichenprogramm”

pgf est un langage de dessin vectoriel bas niveau, qui définit des primitives essentielles pour dessiner. Il est construit comme couche intermédiaire entre l'utilisateur et les *drivers* utilisés par T_EX pour produire les fichiers .dvi, .ps et .pdf.

TikZ est une surcouche de **pgf** qui définit des commandes riches et intuitives pour dessiner. On peut l'étendre avec des modules supplémentaires.

Principaux avantages de TikZ :

- 100% L^AT_EX ! Le document est donc unique, pas de fichiers externes.
- TikZ peut intervenir ailleurs que dans des boîtes d'images : encadrement de textes, entêtes et bas de pages, tableaux, flèches entre éléments du texte (même page)...
- Langage de programmation compatible avec celui de L^AT_EX : familiarité, utilisation dans des commandes L^AT_EX personnelles...
- Nombreuses fonctionnalités de dessin : formes usuelles, couleurs, transparences, dégradés, textures, boucles, conditions, couches...
- Nombreux modules pour des fonctionnalités diverses : matrices, arbres, flèches, décorations, calendriers, chaînes, ombres...
- Installé par défaut avec TeXLive ➡ utilisable partout où il y a L^AT_EX.

La gourmandise de TikZ

TikZ est très gourmand en registres (longueurs, compteurs...).

Lorsqu'on l'installe avec d'autres *packages*, comme par exemple **xypic**, il arrive qu'on sature ce nombre de registres.

On aboutit alors à des messages d'erreurs du type :

```
! No room for a new \dimen
! No room for a new \count
```

Heureusement, le moteur pdf_{te}x (basé sur ϵ -T_EX, une extension de T_EX apparue en 1996) utilisé par défaut aujourd'hui peut allouer beaucoup plus de registres : de $256 = 2^8$ on passe à $32768 = 2^{15}$.

Mais ces registres ne sont pas disponibles par défaut ! Il faut utiliser :

```
\usepackage{etex}
```

juste après `\documentclass[-]{-}` pour activer ces registres supplémentaires.

Il doit être placé avant l'installation d'autres *packages* afin que ces derniers profitent de tous les registres disponibles.

Lorsqu'on utilise TikZ, il est préférable d'utiliser le *package* **etex**.

Cette solution est préconisée dès que le problème de saturation intervient.

La syntaxe de TikZ : les bases

Les figures créées avec TikZ sont contenues soit dans la commande `\tikz{-}`, soit dans l'environnement `tikzpicture`. Pas de différence de rendu entre les deux syntaxes. Cette commande et cet environnement acceptent des options qui seront appliquées à tout le dessin en cours.

```
\usepackage{tikz}
```

```
...
\begin{tikzpicture}[line width=1pt,fill=red,draw=blue]
\path[draw] (0,1)--(1,1)--(1,0);
\filldraw (0,0) circle (0.5cm);
\draw[black] (-5pt,-5pt) rectangle (5pt,5pt);
\end{tikzpicture}
```



```
\tikz[line width=2pt,fill=orange,draw=red]{
\draw[fill] (0,1)--(1,1)--(1,0)--cycle;
\draw[dashed] (0,-0.5) to[out=45,in=135] (1,-0.5);
}
```



TikZ peut être étendu à l'aide de bibliothèques qu'on appelle dans le préambule :

```
\usetikzlibrary{biblio1,biblio2,...}
```

La syntaxe de TikZ : les fondamentaux

Un dessin TikZ obéit aux règles suivantes.

- Les éléments d'un dessin sont des chemins : `\path ... instructions ...` ;
→ `\path` admet des variantes : `\draw`, `\fill`, `\filldraw`, `\node`, `\coordinate`, `\shade`, `\shadedraw`...
- ⚠ La construction d'un chemin se termine avec ;
→ La lecture/construction d'un chemin se fait de gauche à droite.
- Les éléments d'un chemin sont des "opérations" élémentaires qui utilisent le *point courant actuel* et qui définissent le nouveau point courant : aller à un point, insérer une ligne droite, une courbe de Bézier, une forme (rectangle, cercle, ellipse...), un arc, une grille, un graphique...
- En cours de réalisation, le chemin *accumule des options* (`[-]`) : dessiner, remplir, épaisseur du trait, couleurs, style du trait... *Sans option, un chemin est créé sans être rendu*. Le chemin une fois terminé (arrivé au ;) est rendu selon les options données.
- Des nœuds (`node`) peuvent être insérés en cours de réalisation.
Ce sont des morceaux de textes qui seront ajoutés *après* le rendu du chemin.

Les points peuvent être nommés pour les utiliser après.

Il est possible de définir et de nommer des "styles" utilisables dans tous les dessins.

```
\path[draw] (0,0) circle (5pt) to[out=20,in=240] (1cm,2cm) [fill]
node[xshift=10pt] {$A_1$} -| (90:10pt) [thick] -- cycle;
```

Bien se coordonner avec TikZ

TikZ définit différents types de coordonnées.

Coordonnées cartésiennes absolues : ce sont des couples $(\langle dimx \rangle, \langle dimy \rangle)$.

Coordonnées cartésiennes relatives : ce sont des facteurs (fx, fy) , relativement aux unités par défaut définies par $x=1cm$ et $y=1cm$.

Coordonnées polaires : ce sont des couples $(angle:rayon)$, où l'angle est en degrés.

Coordonnées des nœuds : les nœuds peuvent être nommés et il est possible d'utiliser ces noms comme points.

`\coordinate (nom) at (point)` ; définit un tel point.

Coordonnées relatives : `++(coord.)` et `+(coord.)` traduisent le point courant des coordonnées $(coord.)$ (cartésiennes, polaires...).

La première syntaxe redéfinit le point courant, la seconde non.

Expression mathématique : avec la bibliothèque `calc`, on peut écrire des expressions mathématiques du type $(\$f*(point)+g*(point) \dots \$)$.

Coordonnées barycentriques : il est possible de désigner un point comme barycentre d'autres points.

Coordonnées calculées : point de tangence à un chemin, point d'intersection entre deux chemins...

D'autres systèmes de coordonnées peuvent être construits.

Bien se coordonner avec TikZ : exemples

```
\usepackage{tikz}
\usetikzlibrary{calc}
```

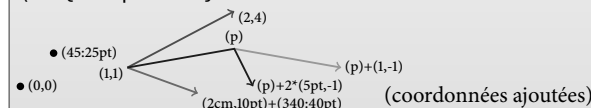
```
...
\begin{tikzpicture}[x=2cm,y=10pt,line width=1pt]
\fill (0,0) circle (2pt);
\fill (45:25pt) circle (2pt);

\draw[red,->] (1,1)--(2,4);

\draw[magenta,->] (2cm,10pt)--++(340:40pt);

\coordinate (p) at (2,2); \draw[orange,->] (p)--+(1,-1);

\draw[blue,->] (1,1)--(p)--($ (p)+2*(5pt,-1) $);
\end{tikzpicture}
```







Faire son chemin avec TikZ : opérations élémentaires

Un chemin commence toujours avec la commande `\path` (ou variante).

Il continue avec une série d'opérations élémentaires utilisant chacune le point courant.

Chaque opération définit un nouveau point courant, en général il s'agit du dernier point mentionné, mais pas toujours.

- `(point)` : le point courant devient le point indiqué.
- `--(point)` : ligne droite entre le point courant et le point indiqué.
- `|-(point)` et `|- (point)` : lignes brisées (horiz.-vert., vert.-horiz.) jusqu'au point.
- `to[-] (point)` définit une ligne (droite par défaut), ses options définissent l'angle de départ et d'arrivée.
- `.. controls (point) and (point) .. (point)` trace une courbe de Bézier.
- `circle[-] (rayon)`, `rectangle[-] (point)`, `ellipse[-]`, `arc[-]`, `sin (point)`, `cos (point)`, `parabola[-] (point)`, `grid[-] (point)`... insèrent des formes.
- Le point courant n'est pas nécessairement modifié.
- `--cycle` ferme un chemin.

```
\draw (0,0)--(1,0.3) (2,0.3)-|(3,0); → 
\draw (0,0) rectangle (12pt,10pt)--(1,0); → 
\draw (0,0) circle (4pt) --(1,0); → 
\draw (0,0) .. controls (1,1) and (2,1) .. (4,0); → 
```

Faire son chemin avec TikZ : options de rendu

- `[draw]` ou `[draw=couleur]` : le chemin est dessiné dans la couleur mentionnée.
- `[fill]` ou `[fill=couleur]` : le chemin est rempli avec la couleur mentionnée. Il faut terminer le chemin par un `--cycle` (sinon TikZ l'ajoute).
- `[color=couleur]` : installe la couleur désignée, elle remplace celles de `draw` et `fill`.
- `[line width=(dim)]` : épaisseur du trait. Des épaisseurs prédéfinies sont fournies : `ultra thin`, `very thin`, `thin`, `semithick`, `thick`, `very thick`, `ultra thick`.
- `[arrows=...]` : définit le type de flèches à insérer au début et à la fin du chemin.
`<-`, `>-`, `0->`, `>>-`, `|>`, `|>-`, `->`, `-<`, `-0`, `->>`, `>->`, `0->`, ...
`[>=...]` : détermine le type de bout de flèche, `to`, `latex`, `stealth`.
La bibliothèque `arrows` ajoute d'autres types de flèches.
- `[double=couleur]`, `[double distance=(dim)]`, ... : définit les traits doubles.
- `[dash pattern=on (dim) off (dim) on (dim) ...]` : définit le type de trait.
Styles prédéfinis : `solid`, `dotted`, `dashed`, `dashdotted`...
- `[rounded corners]` ou `[rounded corners=(dim)]` : les brisures de lignes droites sont arrondies selon le rayon donné (ou un rayon par défaut).
- De nombreuses autres options sont disponibles : `rotate`, `scale`...


La commande `\path` est fondamentale, les autres en dérivent :







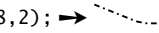
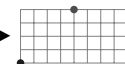
`\draw=\path[draw]`, `\fill=\path[fill]`, `\filldraw=\path[fill,draw]`.

→ Les options passées aux commandes `\draw`, `\fill`, ... sont passées à `\path`.

⚠ Les options les plus récentes écrasent les anciennes de même nature.

Faire son chemin avec TikZ : exemples d'options

```
\path [draw] (0,0) circle (4mm);
\path (0,0) [draw] circle (4mm); → 
\path (0,0) circle (4mm) [draw];
→ On peut écrire la même commande de différentes façons.

\path[fill=black] (0,0) circle (2mm) [fill=red]; → 
\draw[red,line width=3pt] (0,0)--(1,0.3)-|(2,0); → 
\fill[orange] (0,0) rectangle (24pt,12pt); → 
\draw[dotted] (12,0) +(30:10pt) arc (30:95:10pt); →  (centre ajouté)
\draw[blue,line width=2pt,dashed] (0,0)--(1,0.3); → 
\draw[double=white,thick,|>] (0,0)--(10,1); → 
\draw[rounded corners,dashdotted] (0,2)--(5,0)-|(8,2); → 
\draw[help lines] (0,0) grid[step=.25cm] (2,1); → 
\fill[blue] (0,0) circle (2pt);
\fill[red] (1,1) circle (2pt);
```

Un sac de nœuds dans TikZ

Un nœud est un objet graphique qui peut contenir du texte ajouté au chemin courant.

Il est défini lors de la construction du chemin, mais il est dessiné *après* le chemin.

Un nœud a une forme et donc une taille.

La syntaxe générale est la suivante :

```
\path ... node[options] (nom) at (point) {texte} ... ;
```

- Les options `[options]` définissent la forme et donnent les instructions de rendu.
- `(nom)` est optionnel : ce nom peut servir de point nommé utilisable après.
Selon la forme, on peut faire référence à certains points sur le bord du nœud :
`(nom.south)`, `(nom.south east)`, `(nom.north west)`, `(nom.60)` (`60°`)...
- `at (point)` est optionnel : par défaut le nœud est placé au point courant.
Avec cette option, il peut être placé ailleurs.
Un nœud ne change pas le point courant du chemin dans lequel il s'insère.
- `{texte}` est le texte que contient le nœud si la forme (shape) l'accepte.
Pour certaines formes, les accolades sont obligatoires mais peuvent être vides.

TikZ définit les raccourcis suivants : `\node=\path node`,

`\coordinate=\path coordinate=\node[shape=coordinate]`.

```
\draw (0,0)--(2,0) node[anchor=south,circle,draw] {A} --(4,0);
```

→ 

Un sac de nœuds dans TikZ : les options

Les options de nœuds les plus importantes sont :

Forme : `shape=` parmi `circle`, `rectangle`, `coordinate`.

`coordinate` n'accepte pas de texte → crée un point "vide", donc sans taille.

Les bibliothèques `shape.geometric` et `shape.misc` ajoutent d'autres formes.

Rendu : `fill`, `draw`, `rotate`, `scale`...

→ couleurs de rendu, angle de rotation, facteur d'échelle...

Les options usuelles des chemins sont souvent utilisables.

⚠ Certaines options du chemin `ambient` ne sont pas appliquées aux nœuds.

Géométrie : `inner sep`, `outer sep`, `minimum height`, `minimum width`...

→ taille de la forme, marges autour du texte, marges externes...

Positionnement : `anchor=` parmi `south`, `west`, `north`, `east`, `base`, `center`, `east`...

→ point d'ancrage du nœud sur le point courant en fonction de sa forme.

`above=<dim>` équivaut à `anchor=south` avec un décalage vertical de `<dim>`.

→ `below=<dim>`, `right=<dim>`, `left=<dim>`, `above right`, `below left`...

La bibliothèque `positioning` étend ces options.

Texte : `text=<couleur>`, `font=...` sont les attributs du texte;

`align=` parmi `left`, `center`, `right`... pour la mise en forme;

`text width`, `text height`, `text depth`... pour l'aspect général.

Dénouer les nœuds de TikZ

`\node[draw,rectangle,thick,rotate=30] {Texte};` →

`\node[font=\bfseries] at (0,0) {Texte};` → **Texte**

`\fill[blue] (0,0) circle (4pt) node[left=3pt,red]{A};` →

`\node[text width=2cm,align=center,inner sep=0pt,draw] at (0,0) {Texte plus long pour centrer};` →

`\node[top color=blue!10,bottom color=blue!50,rounded corners=5pt,draw=red!80!black,thick] at (0,0) {Texte};` →

`\usetikzlibrary{shapes.symbols}`
`\node[starburst,fill=yellow,draw=red,line width=2pt,inner sep=6pt] at (0,0) {\bfseries Vive \LaTeX{} !};` →

Autres fonctionnalités de TikZ

TikZ admet quelques fonctionnalités de programmation comme les boucles :

`\foreach \r in {1,2,...,5} \draw (0,0) circle (\r mm);` →

`\foreach \x/\t in {1/A,2/B,3/C}{\node[rectangle,draw,above] at (\x,0) {\t}; \draw (\x,0) circle (2pt);}` →

Il est possible de tracer des courbes analytiques de fonctions usuelles :

`\draw[red] plot [domain=-1:1] (\x,{(\x)^2});` →

`\draw plot [domain=-pi:pi] (\x,{sin(\x r)});` →

TikZ permet de définir des "styles" qui résument sous un nom court un lot de réglages.

Ces styles sont utilisables dans plusieurs graphiques.

`\tikzset{rond bleu/.style={draw,color=blue,circle,thick}}` →
`\tikz{\node[rond bleu] at (0,0) {1};}`

Si on n'utilise un style que pour un seul graphique, il est possible de le définir dans la partie optionnelle de `\tikz` ou de `tikzpicture` (sans la commande `\tikzset{-}`).

Les modules externes de TikZ

TikZ est accompagné de nombreux modules externes qui ajoutent des fonctionnalités.

`calc` : Permet de spécifier des coordonnées par des opérations mathématiques.

`($ {1+1}*(1,.5)$), ($\cos(45)*\sin(30)*(1,1)$)`

`intersections` : Permet de lister les points d'intersection entre deux chemins.

`positioning` : Permet de positionner des nœuds les uns relativement aux autres.

`matrix` : Permet de distribuer des nœuds sur un réseau de points.

→ Syntaxe semblable à celle des tableaux L^AT_EX...

`arrows` : Ajoute des types de flèches.

`shapes` : Ajoute de nouvelles formes pour les nœuds (`shape`).

`decorations` : Ajoute des "décorations" au chemins : ondes, ressort, symboles, zigzags...

`plotmarks` : Ajoute des nouveaux types de points pour les graphes.

`shadows` : Ajoute des types d'ombres semi-transparentes.

`shadings` : Ajoute différents types de dégradés de couleurs.

`patterns` : Ajoute des types de motifs pour remplir des zones fermées.

`fit` : Permet d'ajuster un nœud pour qu'il contienne un ensemble de coordonnées.

`external` : Donne la possibilité d'exporter facilement un dessin en un fichier externe.

Voir aussi :

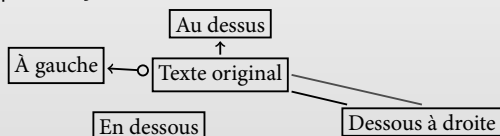
`fading`, `backgrounds`, `circuits`, `automata`, `calendar`, `chains`, `mindmap`...

Illustration du module positioning

```
\usetikzlibrary{positioning}
...
\begin{tikzpicture}[node distance=0.25cm,line width=1pt,
  every node/.style={draw,rectangle,inner sep=3pt,outer sep=2pt}]

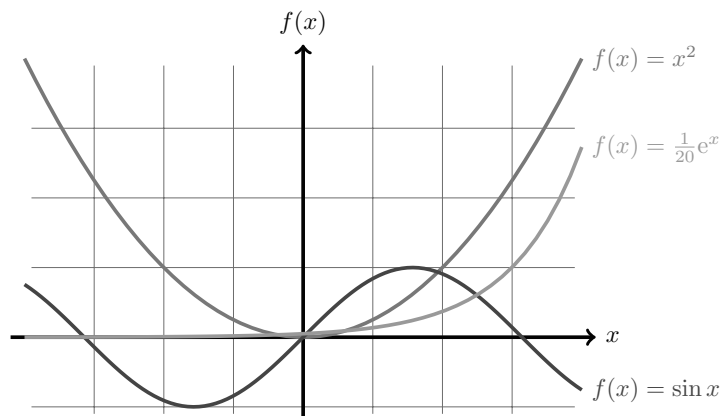
\node (T0) {Texte original};% par défaut: (0,0)
\node[above=of T0] (AD) {Au dessus};
\node[left=of T0,yshift=+5pt,xshift=-15pt] (AG) {À gauche};
\node[below right=of T0,xshift=+20pt] (DD) {Dessous à droite};
\node[below=of T0.south west] (ED) {En dessous};

\draw[->] (T0)-(AD); \draw[o->] (T0)-(AG);
\draw (T0)-(DD); \draw[red] (T0.east)-(DD.north);
\end{tikzpicture}
```



Courbes analytiques dans TikZ avec GNUPlot

Avec l'aide de GNUPlot, il est possible de tracer des courbes analytiques.
TikZ délègue à GNUPlot le soin de préparer un certain nombre de points de la courbe.



Lors d'une première compilation, TikZ génère un fichier .gnuplot qu'il faut faire interpréter par GNUPlot pour produire des fichiers de points.
Une seconde compilation permet à TikZ de lire ces fichiers de données.

Quelques conseils pour réaliser un dessin

À partir d'une idée précise, comment réaliser un dessin programmé ?

Quelques conseils pratiques pour concrétiser ses projets :

1. Dessiner un croquis à main levée sur du papier.
→ En extraire la taille globale évaluée du dessin final.
2. Repérer les points essentiels, les relations entre les objets, les structures mathématiques (courbes, intersections, tangentes, perpendiculaires...)...
→ Cette étape est compliquée mais cruciale, car elle détermine la suite.
3. Créer un fichier L^AT_EX uniquement pour l'image.
→ Compilations rapides et donc fréquentes.
Créer un environnement `tikzpicture` et placer une grille à la taille évaluée :
`\draw[step=.5cm,help lines] (-1,-3) grid (10,3);`
Préférer les coordonnées relatives (1cm).
4. Placer les points essentiels du dessin en les nommant :
`\coordinate (p1) at (2,3);, etc`
→ Les visualiser momentanément avec `\draw (p1) circle (3pt);`
5. Concrétiser les relations structurelles entre les objets, en partant des points nommés.
→ C'est la partie la plus difficile, où la documentation peut se révéler très utile !
6. Commenter les aides au dessin : grille de fond, points de structure...
7. Fignolage : taille globale avec les options `x=...` et `y=...`, épaisseurs de certains traits, ajustement de la position de certains labels, mise en place de styles...

Asymptote ou TikZ ?

La finalité et l'utilisation de ces deux "logiciels" de dessin ne sont pas les mêmes :

- La puissance de calcul de Asymptote est supérieure à celle de TikZ.
→ Opérations de base plus nombreuses, fonctions analytiques, manipulations mathématiques avancées, dessins en 3D, rapidité du rendu...
 - Asymptote produit par défaut des dessins extérieurs.
→ Gain de temps à chaque compilation du document, usages diversifiés...
 - La syntaxe de Asymptote est indépendante de celle de L^AT_EX.
→ Plus proche de langages de programmation courants...
 - TikZ interagit très facilement avec les autres éléments d'un document.
→ Utilisation des mêmes polices d'écriture, création de cadres et de fonds colorés, décorations dans l'entête ou le pied de page, insertion de dessins "dynamiques" dans **beamer**, création de "tableaux" et de diagrammes mathématiques...
 - TikZ définit la notion de style.
→ Séparation du fond sémantique du dessin et de sa forme, code plus léger...
 - TikZ admet plus de possibilités quant au positionnement des éléments entre eux.
→ Inutilité de recourir à des coordonnées exactes, grand choix de points d'ancrage sur des éléments existants, création de diagrammes structurels plus aisée...
- Il faut définir correctement ses besoins pour choisir l'un ou l'autre...

.....
Autres environnements de programmation de dessins

D'autres environnements de programmation de dessins sont disponibles :

L'environnement `picture` : C'est un environnement natif de L^AT_EX qui permet de faire des dessins *très rudimentaires*. Les éléments du dessin sont extraits de fontes spéciales.

Des *packages* étendent ses fonctionnalités : **epic**, **eepic**, **pict2e**, **curve2e**...

⚠ Pour des raisons de compatibilité, c'est `picture` qui est utilisé par **Inkscape** et d'autres logiciels pour superposer une image et ses libellés textuels...

Le *package* **PSTricks** : Programmation de dessin très riche et puissante au sein du document L^AT_EX, report des ordres graphiques sur le *driver*.

Défaut : compilation vers POSTSCRIPT ➔ non compatible avec pdf_{La}TeX.

➔ On lui préférera TikZ, qui est beaucoup plus riche et universel.

Le *package* **xy** : La syntaxe est très compliquée, nombreuses limitations.

➔ à réserver pour les diagrammes mathématiques où il reste efficace.

Le moteur **METAPOST** ¹³¹ : C'est l'ancêtre de Asymptote.

Limitations : précision des calculs en virgule flottante, formats de sorties...

➔ On lui préférera Asymptote qui le remplace avantageusement.

Convertir du code METAPOST en Asymptote n'est pas un gros travail.

Logiciels externes au monde L^AT_EX : De nombreux langages de programmation proposent des modules de dessins programmés.

Par exemple **Matplotlib** ¹³² permet de dessiner dans le langage PYTHON ¹³³.

7.5 Insertion de graphiques externes

.....
La commande `\includegraphics[-]{-}` de **graphicx**

C'est la commande la plus importante du *package* **graphicx**.

Cette commande insère une image dans le flot courant, en tant que boîte.

L'argument obligatoire est le nom d'un fichier d'image, avec éventuellement le chemin vers un sous dossier, l'argument optionnel définit des options sous la forme d'une liste d'éléments du type

`clé=valeur` :

`\includegraphics[width=2cm]{image.png}`

`\includegraphics[angle=45,origin=c]{dossier/image.pdf}`

Les clés principales et les plus utiles sont les suivantes :

Clé	Description	Exemple
<code>width</code>	largeur de l'image	<code>width=1cm</code>
<code>height</code>	hauteur de l'image	<code>height=50pt</code>
<code>scale</code>	mise à l'échelle de l'image	<code>scale=2</code>
<code>angle</code>	angle de rotation	<code>angle=45</code>
<code>origin</code>	origine de la rotation	<code>origin=br</code>
<code>viewport</code>	définit la zone à afficher (bp par défaut)	<code>viewport=0 0 72 72</code>
<code>trim</code>	définit les marges à enlever	<code>trim=10 15 5 7</code>
<code>clip</code>	réduit la zone à afficher	<code>clip=true</code>

D'autres clés sont décrites dans la documentation du *package* **graphicx**.

.....
La commande `\includegraphics[-]{-}` : exemples

`\includegraphics[width=1.7cm]{pomme.png}`



`\includegraphics[width=1.2cm]{dos.png}`

`\includegraphics[width=1.2cm,origin=c,angle=45]{dos.png}`

`\includegraphics[origin=c,angle=-45,width=1.2cm]{dos.png}`



(cadres et lignes ajoutés)

➔ L'ordre des options est important.

➔ Une figure sans rotation n'a pas de profondeur, une figure tournée peut en avoir une.

`\includegraphics[viewport=4.6cm 1.9cm 5.1cm 2.4cm,
clip,
height=1cm]{CPT.pdf}`



Autres commandes utiles du `package graphicx`

La commande `\graphicspath{-}` permet de spécifier une liste de dossiers où L^AT_EX peut rechercher des images :

```
\graphicspath{{../pdf/}{png/}}
```

La syntaxe est celle des chemins UNIX.
Rappelons que L^AT_EX cherche aussi les images dans les dossiers `texmf`.

La commande `\DeclareGraphicsExtensions{-}` permet de désigner une liste d'extensions de fichiers :

```
\DeclareGraphicsExtensions{pdf,png,jpg}
```

→ On peut alors écrire `\includegraphics[width=1.2cm]{image}`
et L^AT_EX cherche dans l'ordre `image.pdf`, puis `image.png` et `image.jpg`.



```
\includegraphics[angle=-40]{dahu} →
```

Rotations avec le `package graphicx`

La commande `\rotatebox[-]{-}{-}` permet de tourner tout objet géré par L^AT_EX : graphiques, textes, tableaux...
Le premier argument obligatoire est l'angle (en degrés, sens trigonométrique), le second argument est l'objet à tourner.
L'argument optionnel définit des options sous la forme d'une liste de `clé=valeur` :

Clé	Description	Exemple
<code>origin</code>	origine de la rotation (<code>lrctbB</code>)	<code>origin=br</code>
<code>x</code>	abscisse du centre de rotation	<code>x=50pt</code>
<code>y</code>	ordonnée du centre de rotation	<code>y=5mm</code>
<code>units</code>	changement d'unité de rotation	<code>units=6.283185</code>

`l` = *left*, `r` = *right*, `c` = *center*, `t` = *top*, `b` = *bottom*, `B` = *Baseline*.
`units=6.283185` ($\simeq 2\pi$) permet de donner l'angle en radians (défaut : `units=360`).
"angle final en degrés, sens trigo." = $\frac{360}{\text{units}} \times$ "valeur donnée".

```
ABCD
\rotatebox[origin=c]{45}{ABCD}
\rotatebox[origin=tr]{90}{ABCD}
\rotatebox[x=0pt,y=15pt]{60}{ABCD}
\rotatebox[origin=c,units=-360]{45}{ABCD}
```

Mises à l'échelle par un facteur

Les commandes suivantes du `package graphicx` permettent d'appliquer des mises à l'échelle sur tout objet géré par L^AT_EX : graphiques, textes, tableaux...

La commande `\scalebox{H}[V]{texte}` met à l'échelle *texte* : facteur *H* dans la direction horizontale, facteur (optionnel) *V* dans la direction verticale.
Si le facteur *V* n'est pas spécifié, la mise à l'échelle se fait avec *H* dans les deux directions.

```
ABCD  ABCD
\scalebox{2}[0.5]{ABCD}  ABCD
\scalebox{2}{ABCD}  ABCD
\scalebox{1}[-2]{ABCD}  VBCD
```

`\reflectbox{texte}` équivaut à `\scalebox{-1}[1]{texte}`

```
ABCD  ABCD
\reflectbox{ABCD}  DCBA
```

→ Utile pour insérer des textes de Léonard de Vinci ¹³⁴.

Mises à l'échelle vers des dimensions données

La commande `\resizebox{<dim-h>}{<dim-v>}{texte}` du `package graphicx` met à l'échelle *texte* pour atteindre une boîte à la dimension horizontale *<dim-h>* et à la dimension verticale *<dim-v>*.
Si *<dim-h>* ou *<dim-v>* vaut !, l'autre dimension impose la mise à l'échelle en préservant les proportions de *texte*.
On peut utiliser `\height`, `\width`, `\totalheight`, `\depth` dans ces arguments pour faire référence aux dimensions naturelles de *texte*.

```
ABCD  ABCD
\resizebox{1cm}{0.5cm}{ABCD}  ABCD
\resizebox{2.5cm}{!}{ABCD}  ABCD
\resizebox{!}{15pt}{ABCD}  ABCD
\resizebox{\width}{15pt}{ABCD}  ABCD
\resizebox{\width}{3\height}{ABCD}  ABCD
```


Le package pdfpages

Et si on veut insérer une page entière ? Le package **pdfpages** répond à ce problème.

La commande `\includepdf[célé=valeur]{fichier.pdf}` insère certaines pages du fichier PDF *fichier.pdf* dans le document courant, *en tant que pages*.

célé=valeur sert à fournir les options :

- `pages={1, 3, {}, 10-15}` insère les pages 1 et 3 du fichier, puis une page blanche, puis les pages 10 à 15. `pages={5-}` insère toutes les pages à partir de la 5.
- `pages={-}` insère tout le document.
- `nup=2x1` place 2 pages côte à côte sur une page finale.
⚠ Il vaut mieux alors utiliser l'option `landscape` avec le **geometry**.
- `frame` ajoute un cadre autour de chaque page insérée,
`delta=5pt 10pt` ajoute des espaces horizontaux et verticaux entre les pages.
- D'autres options permettent de disposer les pages insérées de nombreuses façons : `openright`, `landscape`, `offset`, `column`, `signature`...

➔ Consulter la documentation du package **pdfpages**.

```
\usepackage[a4paper,margin=1cm,landscape]{geometry}
\usepackage{pdfpages}
...
\includepdf[pages={-}, nup=2x1, openright]{document.pdf}
```

➔ permet de préparer un document pour imprimer deux pages sur une page physique.

7.6 Insertion d'une figure avec légende

Rappels sur l'environnement figure

`\includegraphics[-]{-}` se contente d'insérer une image en tant que boîte.

➔ l'image est placée à l'endroit où la commande est appelée, comme s'il s'agissait d'un simple (gros) caractère.

Une figure est souvent plus qu'une simple image : légende, numéro pour y faire référence, disposition centrée dans le texte... L'environnement `figure` de L^AT_EX sert à :

- définir une figure comme un flottant
➔ disposition dans le flux du texte à un endroit approprié ;
- insertion d'une légende avec numérotation automatique ;
- possibilité de faire référence à la figure grâce à un label.

```
\begin{figure}[ht]
\centering
\includegraphics[width=0.5\textwidth]{image.png}
\caption{Une légende pour décrire la figure}
\label{fig-image}
\end{figure}
```

La commande `\label{-}` doit être placée après la commande `\caption{-}` puisqu'elle fait référence au numéro de la figure créé par cette dernière.

Voir le cours sur les flottants pour gérer cet aspect de l'environnement `figure`.

⚠ N'utiliser `figure` que si on a besoin d'une légende et/ou d'un numéro...

Personnaliser l'environnement figure

Quelques dimensions et commandes sont accessibles pour personnaliser l'aspect d'une figure, et plus généralement d'un flottant :

- Les longueurs `\abovecaptionskip` et `\belowcaptionskip` gèrent les espaces verticaux avant et après la légende.
➔ Utile lorsqu'on place la légende au-dessus de l'image.
- La commande `\topfigrule` est exécutée après la dernière figure *top* d'une page pour séparer les figures du texte.

La commande `\botfigrule` fait de même avant la première figure *bottom*.

⚠ Par défaut, ces commandes sont "vides" : elles se modifient (la première fois) avec `\newcommand{-}{-}` et la hauteur finale doit être nulle.

```
\newcommand{\topfigrule}{%
\vspace*{10pt}\hrule\vspace{-10.4pt}}
```

- `\floatsep` est l'espacement vertical entre les flottants *top* ou *bottom*.
- `\intextsep` est l'espacement vertical avant et après un flottant de type `h`.
- `\textfloatsep` est l'espacement vertical entre les flottants et le texte.

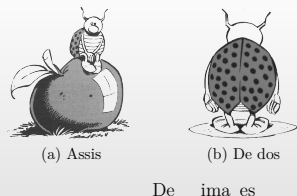
Le document **Using imported Graphics in L^AT_EX and pdfL^AT_EX**¹³⁵ donne de nombreux trucs et astuces concernant le positionnement de graphiques dans le texte.

⚠ Ce texte comporte quelques erreurs, il est donc utile de tester ce qu'il énonce.

Le package subfig

Le package **subfig** place dans un même flottant plusieurs “sous flottants” avec leur sous numérotations et des sous légendes.
→ fonctionne aussi avec l'environnement `table`.

```
\begin{figure}[t]
\centering
\subfloat[Assis]{
\includegraphics[height=3cm]{pomme.png}
\label{fig-pomme}}
\subfloat[De dos]{
\includegraphics[height=3cm]{dos.png}
\label{fig-dos}}
\caption{Deux images}\label{fig-double}
\end{figure}
```



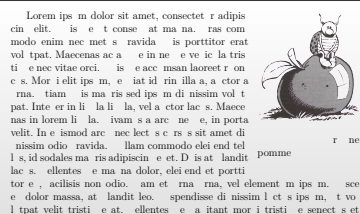
Ce package permet de lister les sous flottants, de modifier l'aspect et le positionnement des légendes...

⚠ Conflit de noms : La commande `\subfloat` n'a rien à voir avec le package **subfloat** qui définit les environnement `subfigures` (!) et `subtables` (dans le même esprit que `subequations`) pour modifier les numérotations des figures et tableaux.

Le package wrapfig

Le package **wrapfig** permet d'insérer une figure dans un paragraphe.

```
\begin{wrapfigure}{r}{4cm}
\includegraphics[width=4cm]{pomme.png}
\caption{Sur une pomme}
\end{wrapfigure}
Lorem ipsum dolor sit amet, ...
```



L'insertion peut se faire à droite, à gauche, sur le côté intérieur ou le côté extérieur dans le mode recto-verso (`twoside`).

L'image peut déborder dans la marge, l'espacement entre le texte et l'image est modifiable.

Ce package est déjà ancien, mais il fonctionne toujours aussi bien.

Par contre, sa documentation n'est pas fournie en PDF.

`texdoc wrapfig` affiche le fichier `wrapfig.sty`...

Les packages caption et float

Il est possible de personnaliser les légendes des flottants avec le package **caption**.

- Une commande `\captionsetup[-]{-}` permet de modifier de façon globale l'aspect des légendes d'un type particulier de flottants.
- Il est possible de modifier le format général de la légende, la typographie des éléments d'une légende, les espacements verticaux et horizontaux...
- Des styles peuvent être définies et chargés.

Il est possible de définir de nouveaux types de flottants avec le package **float**.

- Une commande `\newfloat` permet de définir de nouveaux types de flottants avec des styles prédéfinis.
→ commande dans le même esprit que `\newtheorem`.
- Une liste des flottants ainsi définis est accessible.
- Ce package a déjà été mentionné car il définit l'option de placement `[H]`.

Consulter la documentation de ces packages pour plus de détails et d'exemples.

Ces packages sont compatibles avec les packages présentés auparavant.

Cours 8 – Gérer la bibliographie



8.1 La gestion manuelle de la bibliographie	156
Où l'on s'obstine à (re)copier et (re)coller inlassablement des entrées bibliographiques à la fin de nos documents...	
8.2 Une base de données bibliographiques	157
Où l'on découvre une solution plus performante et plus pratique pour gérer sa bibliographie.	
8.3 Personnaliser les citations	165
Où l'on découvre quelques modules pour modifier l'aspect des citations bibliographiques dans notre document...	
8.4 Gérer sa base bibliographique	170
Où l'on se rend compte qu'un bon logiciel dédié vaut toujours mieux qu'un simple traitement de texte...	
8.5 L'avenir avec bib_latex	172
Où l'on pose un pied dans un avenir plus radieux mais pas encore tout à fait opérationnel...	

8.1 La gestion manuelle de la bibliographie

L'environnement thebibliography

L'environnement thebibliography débute une section non numérotée (**article**) ou un chapitre non numéroté (**book**), et installe un environnement de liste.

```
\begin{thebibliography}{AAAA}
...
\bibitem[Ber96]{Bert96}
  R.~Bertlmann.
  \newblock {\em Anomalies in Quantum Field Theory}.
  \newblock Oxford Science Publications, 1996.
...
\end{thebibliography}
```

L'argument de l'environnement thebibliography sert à fournir la largeur de la plus grande clé apparente.

Chaque item est nommé \bibitem[-]{-} : l'argument optionnel est la clé apparente dans le document. Sans cette option, chaque \bibitem produit un numéro.

On peut mélanger des clés manuelles et des clés automatiques.

L'argument obligatoire de \bibitem est la clé interne de citation (→ \cite{-}).

Les commandes \newblock (option) permettent de séparer les entités sémantiques.

Selon la classe du document, elle peut ajouter un espace ou passer à la ligne...

Les limitations d'une bibliographie manuelle

Que peut-on personnaliser dans une bibliographie manuelle ?

- On peut redéfinir le titre de la section ou du chapitre des références :
`\renewcommand{\refname}{Liste de livres}`
- On peut formater comme on le souhaite les entrées bibliographiques.
 → Énorme travail pour une bibliographie conséquente...
- On peut modifier le comportement de \newblock si on en a introduit.
 Dans les classes usuelles (**article**, **book**) cette commande est définie comme :
`\newcommand\newblock{\hspace{.11em}\@plus.33em\@minus.07em}`
 → espace horizontal de type ressort.
 On peut lui demander de couper les lignes :
`\renewcommand{\newblock}{\newline}`

La commande \cite[-]{-} permet de citer une entrée bibliographique.

L'argument obligatoire est la clé interne de citation, l'argument optionnel permet d'ajouter du texte, comme par exemple un numéro de page, un numéro de théorème...

`\cite[p. 100]{Bert96}` → [Ber96, p. 100]

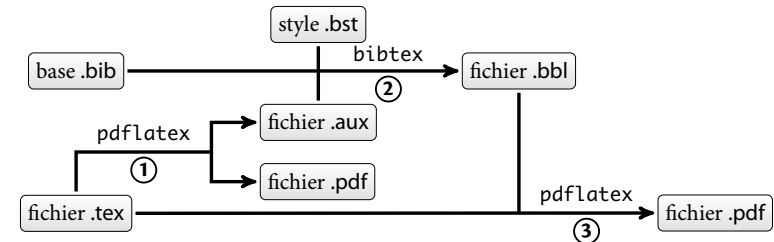
Les problèmes évidents d'une bibliographie gérée manuellement : changer l'ordre des entrées, reformater les entrées, supprimer une citation et l'entrée correspondante...

Gérer des entrées bibliographiques à la main (copier-coller) n'est pas une solution pérenne.

8.2 Une base de données bibliographiques

L'utilisation d'une base de données

Le moteur bibtex¹³⁶ permet à L^AT_EX d'utiliser une base de données bibliographiques centralisée dans un fichier .bib.



1. La première compilation insère des informations dans le fichier .aux à partir des commandes \cite{-} (clés de citations internes).
2. Le moteur bibtex utilise ces informations pour extraire les données bibliographiques du fichier .bib, il les formate selon un fichier de style .bst et il crée un fichier .bbl.
3. La seconde compilation insère le fichier de bibliographie formaté .bbl.

En général, un document se stabilise avec la séquence suivante de compilations :

pdflatex → bibtex → pdflatex → pdflatex

Mise au point

Une mise au point est nécessaire pour comprendre le rôle des composants :

- Les fichiers .bib constituent la base de données des références bibliographiques.
 → La base peut être morcelée en plusieurs fichiers.
 Dans ces fichiers on y trouve des “fiches” décrivant chaque entrée bibliographique. Ces fichiers peuvent contenir des centaines de fiches bibliographiques. Des logiciels permettent de gérer facilement cette base de données.
 Rappel : les fichiers .bib se placent dans un dossier texmf/bibtex/bib/.
- Les fichiers .bst sont des fichiers de style.
 En réalité ce sont des scripts que le moteur bibtex interprète pour formater les entrées bibliographiques.
 Le langage de script de ces fichiers de styles est assez compliqué, d'où la difficulté à personnaliser un fichier de style.
 Un simple changement de fichier de style peut modifier l'apparence des entrées bibliographiques (formatage et ordre) et aussi des clés de citations externes (visibles).
- Le fichier .bbl contient en réalité un simple environnement thebibliography préparé par bibtex à l'aide de quelques entrées extraites des fichiers .bib et des instructions du fichier de style .bst choisi.

On pourra consulter le document PDF intitulé **Tame the Beast**¹³⁷ pour de la documentation sur bibtex.

.....

Le code dans le document source

Pour utiliser `bibtex` il faut donc renseigner :

Le ou les fichiers de données bibliographiques : Pour cela on insère une ligne à l'endroit où on veut que la bibliographie apparaisse dans le document :

`\bibliography{fichierun,fichierdeux,...}` (sans l'extension `.bib`)

→ On peut donc découper sa base bibliographique en plusieurs fichiers `.bib`.

⚠ Une clé de citation interne ne peut cependant pas être présente dans deux fichiers utilisés en même temps dans un document !

Le fichier de style : Il faut désigner un fichier de style `.bst` ("*bibtex style*") :

`\bibliographystyle{unsrt}` (sans l'extension `.bst`)

Cette ligne peut-être insérée n'importe où, par exemple dans le préambule.

Chaque commande `\cite{-}`, avec une clé interne faisant référence à une entrée de la base de données, insère l'entrée bibliographique correspondante dans la bibliographie.

`\nocite{-}` permet d'inclure une référence bibliographique sans la citer.

`\cite{*}` et `\nocite{*}` permettent d'inclure toutes les références bibliographiques de la base de données (ou des bases de données) utilisée.

.....

Le moteur `bibtex`

`pdflatex` (ou `latex`) insère toutes les informations qui sont utiles à `bibtex` dans le fichier `.aux` : les clés internes des entrées bibliographiques utilisées, les fichiers de base de données, et la feuille de style.

→ Il faut lancer `bibtex` sur le fichier `.aux`.

→ Si des citations sont ajoutées ou ôtées du texte, il faut relancer `bibtex` pour mettre à jour le fichier `.bbl`.

Selon le fichier de style, `bibtex` peut être amené à effectuer des classements d'entrées en ordre alphabétique (auteurs, éditeurs...).

Ceci peut poser des problèmes avec les lettres accentuées, et avec certaines langues.

→ C'est un des points faibles de `bibtex`.

`bibtex8` est un moteur qui prolonge `bibtex` :

– Il admet une capacité mémoire supérieure :

`bibtex8 -W fichier.aux` lui donne les capacités maximales.

– Il accepte des encodages sur 8 bits.

→ On peut lui désigner un fichier expliquant l'ordre alphabétique choisi.

`bibtex8 -c 88591lat.csf fichier.aux`

impose l'encodage ISO 8859-1⁷⁸ sur 8 bits.

→ pour des articles en anglais, `bibtex` suffit largement.

.....

Les entrées de la base : exemples

Exemples d'entrées bibliographiques dans un fichier `.bib` :

```
@book{BoosBlee85,
  Author = {Booss, B. and Bleecker, D. D.},
  Publisher = {Springer-Verlag},
  Series = {Universitext},
  Title = {Topology and Analysis, The {A}ttiyah-{S}inger Index
          Formula and Gauge-Theoretic Physics},
  Year = {1985}
}

@article{t-HoVelt72a,
  Author = {'t Hooft, G. and Veltman, M.},
  Journal = {Nucl. Phys. B},
  Pages = {189--213},
  Title = {Regularization and renormalization of gauge fields},
  Volume = {44},
  Year = {1972}
}
```

.....

Les entrées de la base : généralités

Chaque entrée est structurée de la façon suivante :

Un type : qui désigne le type d'entrée bibliographique : livre, article, conférence...

`@book{...}, @article{...}`

→ les accolades encadrent toute l'entrée bibliographique.

La clé de citation interne : c'est la clé qui sera utilisée pour retrouver l'entrée dans la base à travers les commandes `\cite{-}`.

Caractères acceptés : lettres non accentuées, chiffres, tirets, doubles points...

Pas d'espace, pas de @...

→ en général, établir une règle personnelle et s'y tenir : mentionner les auteurs et l'année par exemple.

`BoosBlee85, t-HoVelt72a, Bert96...`

Une série de mots-clés et de valeurs : ils renseignent l'entrée bibliographique, ils sont séparés par des virgules, les valeurs sont encadrées par des accolades (ou "...").

Les valeurs sont codées en `LaTeX` : mathématiques, tirets longs (`--` → `-`)...

`Author = {'t {H}ooft, G. and Veltman, M.},`

`Journal = {Nucl. Phys. B},`

`Pages = {189--213},`

`bibtex` est insensible à la casse concernant les mots de son langage :

`@article = @Article = @ARTICLE, author = Author = AUTHOR, etc.`

.....

Les entrées de la base : les types

Il existe de nombreux types d'entrées bibliographiques.

Chaque type d'entrée définit des mots-clés obligatoires et des mots-clés optionnels.

Il est possible d'ajouter des mots-clés personnels : ils seront ignorés par les styles habituels.

➔ Ils peuvent être utiles pour gérer d'autres métadonnées de la base bibliographique.

@article : article d'un journal ou d'un magazine.

obligatoire : author, title, journal, year.

optionnel : volume, number, pages, month, note.

@book : un livre avec un éditeur.

obligatoire : author ou editor, title, publisher, year.

optionnel : volume ou number, series, address, edition, month, note.

@inproceedings : article dans un compte-rendu de conférence.

obligatoire : author, title, booktitle, year.

optionnel : editor, volume ou number, series, pages, address, month, organization, publisher, note.

@unpublished : un document ayant des auteurs et un titre mais non publié.

obligatoire : author, title, note.

optionnel : month, year.

.....

Les entrées de la base : les types (suite)

@proceedings : compte-rendu de conférence.

obligatoire : title, year.

optionnel : editor, volume ou number, series, address, publisher, note, month, organization.

@inbook : partie d'un livre.

obligatoire : author ou editor, title, chapter et/ou pages, publisher, year.

optionnel : volume, number, series, type, address, edition, month, note.

@incollection : partie d'un livre ayant son propre titre.

obligatoire : author, title, booktitle, publisher, year.

optionnel : editor, volume ou number, series, type, chapter, pages, address, edition, month, note.

@booklet : livre sans éditeur.

obligatoire : title.

optionnel : author, howpublished, address, month, year, note.

.....

Les entrées de la base : les types (suite)

@masterthesis : thèse de fin d'étude.

obligatoire : author, title, school, year.

optionnel : type, address, month, note.

@phdthesis : thèse de doctorat.

obligatoire : author, title, school, year.

optionnel : type, address, month, note.

@manual : documentation technique.

obligatoire : title.

optionnel : author, organization, address, edition, month, year, note.

@techreport : un rapport publié par une institution.

obligatoire : author, title, institution, year.

optionnel : type, number, address, month, note.

@misc : ce qui ne rentre pas dans les autres types.

obligatoire : aucun.

optionnel : author, title, howpublished, month, year, note.

.....

Les entrées de la base : les auteurs

Les mots clés author et editor doivent être remplis avec des règles précises.

Chaque auteur/éditeur est séparé des autres par and.

Pour trier au mieux les références bibliographiques, le moteur bibtex doit être capable de faire la différence entre les *prénoms*, le *nom*, les *particules* et le *complément*.

Il existe plusieurs règles possibles pour différencier ces 4 éléments :

Prénoms Nom Prénoms particules Nom particules Nom, Prénoms

particules Nom, Compléments, Prénoms

Les majuscules sont importantes pour reconnaître les éléments.

– Paul Adrien Maurice Dirac = Dirac, Paul Adrien Maurice

➔ "Dirac" est le nom, le reste désigne les prénoms.

– John von Neumann = von Neumann, John

➔ "von" est la particule, "Neumann" est le nom.

– Alfredo Miguel Fernandez ≠ Miguel Fernandez, Alfredo

➔ la seconde forme est plus précise.

– Smith, Jr., John

➔ "Jr." est le complément.

Utiliser systématiquement la forme : particule Nom, Complément, Prénom.

Les entrées de la base : autres particularités

En fonction du fichier de style choisi, le moteur `bibtex` peut effectuer une mise en minuscule ou en majuscules de tous les mots ou des premières lettres de certains mots des titres ou des autres champs textuels.

Exemple : les titres des livres sont souvent en lettres capitales, mais pas les titres des articles.

Lorsqu'on a besoin de préserver le formatage, on peut placer des accolades :

```
Title = {Topology and Analysis, The {A}tiyah-{S}inger Index
        Formula and Gauge-Theoretic Physics}
```

```
Title = {An Invitation to {$C^{\ast}$}-Algebras}
```

Les mots “Atiyah” et “Singer” conserveront toujours leur première lettre en majuscule, le “C” de “C*-Algebras” restera en majuscule.

Les lettres accentuées sont de préférences encodées sur 7 bits, *entre accolades* :

```
G{"o}del, Kurt.
```

Pour les numéros de pages, `bibtex` ne formate pas le champ.

On peut donc le remplir directement avec ce qui sera présenté dans la bibliographie.

Rappel : un tiret long exprimant un intervalle de page se formate `n--m` plutôt que `n-m`.

Les entrées de la base : références croisées

Il est possible de créer des références croisées dans la base.

```
@inbook{Cunt04a,
  Author = {Cuntz, Joachim},
  Crossref = {CuntSkanTsyg04},
  Pages = {1--71},
  Title = {Cyclic Theory, Bivariant {$K$}-Theory and
    the Bivariant {C}hern-{C}onnes Character}}
```

```
@book{CuntSkanTsyg04,
  Author = {Cuntz, Joachim and Skandalis, Georges and
    Tsygan, Boris},
  Publisher = {Springer-Verlag},
  Series = {Encyclopaedia of Mathematical Sciences},
  Title = {Cyclic Homology in Non-Commutative Geometry},
  Volume = {121},
  Year = {2004}}
```

La première entrée fait référence à la seconde.

➔ Les champs qui manquent dans la première entrée sont pris dans la seconde.

Il faut placer les entrées ainsi référencées à la fin de la base de données `.bib`.

Autres informations dans une base de données

On peut définir des raccourcis textuels dans la base `.bib`, en général au début du fichier.

```
@string{EMS = "Encyclopaedia of Mathematical Sciences"}
```

Dans une entrée bibliographique, il est possible d'écrire directement :

```
Type = EMS ou Type = ems
```

△ sans guillemets et sans accolades, insensible à la casse.

On peut enchaîner des chaînes de caractères avec le symbole `#` :

```
@string{PR = "Physical Review"}
```

```
Journal = PR # "~D"
```

```
Journal = PR # "~C"
```

Presque tous les fichiers de style contiennent des abréviations :

- Quelques titres de revues en science informatique.
- Les mois de l'année : jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec.
➔ Il est préférable de les utiliser pour d'éventuelles traductions.

Il est possible de demander à `bibtex` d'inclure du code supplémentaire dans le fichier `.bbl`.

```
@preamble{"\providecommand{\url}[1]{\texttt{\#1}}"
```

➔ la commande `\url{-}` sera accessible aux entrées bibliographiques.

Remarque : La commande `\providecommand` se comporte comme `\newcommand` mais elle vérifie que la commande à définir n'existe pas avant de la définir.

Ceci évite les messages du type “Command ... already defined”.

Les fichiers de style `.bst`

Il existe de nombreux fichiers de style utilisables avec le moteur `bibtex`.

Des revues proposent souvent un fichier de style `.bst` avec leur feuille de style `LATEX`.

Rappel : les fichiers `.bst` se placent dans un dossier `texmf/bibtex/bst/`.

Les fichiers de style principaux sont les suivants :

`plain.bst` : Les entrées bibliographiques sont triées par ordre alphabétique des noms des auteurs et présentées sous la forme `Prénoms Nom`.

Les clés externes sont des numéros.

`alpha.bst` : *idem* que `plain.bst` sauf que les clés externes sont les initiales ou les abréviations des auteurs et l'année.

`abbrv.bst` : *idem* que `plain.bst` sauf que les prénoms sont remplacés par les initiales.

`unsrt.bst` : *idem* que `plain.bst` sauf que les entrées apparaissent dans l'ordre où elles sont citées dans le texte.

`apalike.bst` : Les entrées bibliographiques sont triées par ordre alphabétique des noms des auteurs et présentées sous la forme `Nom, P..`

Les clés externes sont les noms des auteurs et l'année.

Les styles usuels ont été adaptés au français : `plain-fr.bst`, `alpha-fr.bst`...

`texmf-dist/bibtex/bst/` contient l'essentiel des fichiers de style installés dans `TeXLive`¹¹.

Le site **BibTeX Style Examples**¹³⁸ fournit des exemples de rendu des styles usuels.

Transférer un document L^AT_EX avec sa bibliographie

Lorsqu'on doit transférer/échanger un document L^AT_EX contenant une bibliographie gérée par bib_tex, il faut prévoir :

- Le document source .tex.
→ Évident !
- Le fichier .bbl (et lui seul) si on est certain que le formatage des entrées bibliographiques est définitif.
→ Vers une revue si on a utilisé le fichier de style .bst fourni.
→ Vers les serveurs en ligne, comme **arXiv**⁹⁰.

Dans d'autres circonstances :

- Il est conseillé de joindre un extrait de la base de données dans un fichier .bib.
→ Requis si le document n'est pas finalisé : permet des ajouts.
→ Situations où on travaille à plusieurs sur le même document.
→ Penser alors à désigner ce fichier dans \bibliography{-}.
- Si besoin joindre le fichier de style .bst s'il n'est pas dans une distribution standard.

Rappel : les fichiers .bbl, .bib, .bst sont recherchés d'abord dans le dossier de travail.

→ Les placer avec le document .tex.

8.3 Personnaliser les citations

Le package cite

Le package **cite** permet de personnaliser sur de nombreux points le comportement de la commande \cite[-]{-} :

- modification du matériel placé avant la clé de citation, après la clé de citation et entre des clés de citation ;
- mise en ordre numérique et compression des clés numériques successives ([4,9,3,7,5,2] → [2-5,7,9]) ;
- citations numériques sous forme d'exposant ([2] → ²) ;
- ajout d'espaces entre les clés de citations et autres petits détails...

Les options principales du package **cite** :

nosort : Les clés numérotées dans une même commande \cite{-} ne sont pas remises en ordre croissant.

nocompress : Les clés numérotées successives dans une même commande \cite{-} ne sont pas compressées.

super = superscript : Les clés de citations dans le texte sont mises sous forme d'exposants.

biblabe1 : Les clés de citations sont mises en exposant dans la liste bibliographique.

→ à combiner avec **super**.

Le package **cite** fonctionne en mode manuel ou avec bib_tex.

Le package cite : changer l'aspect des citations

\citeleft et \citeright représentent le matériel à gauche et à droite des citations.

Elles sont inopérantes avec l'option **super**.

\citepunct représente le matériel entre les clés d'une même commande \cite{-}.

\citedash représente le matériel entre les clés compressées.

```
\renewcommand{\citeleft}{\{ }
\renewcommand{\citeright}{\} }
\renewcommand{\citepunct}{/\nolinebreak}
\renewcommand{\citedash}{\cdots}
```

[2-5,7] → (2...5/7)

\citemid représente le matériel entre la clé et le texte optionnel.

```
\renewcommand{\citemid}{; \ }
```

[5, p. 10] → [5; p. 10]

\citeform{-} formate le numéro de la citation.

```
\renewcommand{\citeform}[1]{\textbf{\#1}}
```

[5,7] → [5,7]

\citen{-} n'imprime que les clés de citation sans le matériel à gauche et à droite.

Le package natbib

Le package **natbib** permet de gérer les citations numériques à la manière de **cite** et de créer des citations de type “auteurs-année”.

C’est l’un des *packages* les plus complets pour gérer les citations.

Il fonctionne avec une bibliographie manuelle, mais il est préférable de le combiner à **bibtex** pour en profiter pleinement.

➔ Dans ce qui suit on n’expliquera l’utilisation de **natbib** qu’avec **bibtex**.

Trois fichiers de style sont fournis : **plainnat.bst**, **abbrvnat.bst** et **unsrnat.bst** qui reprennent les caractéristiques des fichiers originaux.

Ils ajoutent quelques champs supplémentaires : **isbn** (livres), **issn** (périodiques), **url**, **doi** (*Digital Object Identifier*), **eid** (*Electronic ID*).

natbib définit un certain nombre de commandes de citations utiles surtout dans le schéma “auteurs-année” :

- `\citet[-]{-}` correspond aux citations “textuelles”, sans parenthèses globales.
- `\citep[-]{-}` ajoute des parenthèses globales.
- `\citeyear{-}` donne l’année seulement.
- `\citeauthor{-}` donne les auteurs sans l’année.

Plusieurs clés peuvent être associées dans `\citet` ou `\citep`.

Des versions étoilées présentent *tous* les auteurs (`\citet*`, `\citep*`...).

`\cite[-]{-}` correspond à `\citet[-]{-}`.

Le package natbib : exemples

On se place dans le schéma “auteurs-année” :

<code>\citet{Dixm64}</code>	Dixmier (1964)
<code>\citet[p.~100]{Dixm64}</code>	Dixmier (1964, p. 100)
<code>\citep{Dixm64}</code>	(Dixmier, 1964)
<code>\citep[p.~100]{Dixm64}</code>	(Dixmier, 1964, p. 100)
<code>\citep{Dieu94,Dixm64}</code>	(Dieudonné, 1994; Dixmier, 1964)
<code>\citep{Cart51a,Cart51b,Dixm64}</code>	(Cartan, 1951a,b; Dixmier, 1964)
<code>\cite{BodeHambNicaPark05}</code>	Boden et al. (2005)
<code>\citet{BodeHambNicaPark05}</code>	Boden et al. (2005)
<code>\citep{BodeHambNicaPark05}</code>	(Boden et al., 2005)
<code>\citet*{BodeHambNicaPark05}</code>	Boden, Hambleton, Nicas, and Park (2005)
<code>\citeauthor{BodeHambNicaPark05}</code>	Boden et al.
<code>\citeauthor*{BodeHambNicaPark05}</code>	Boden, Hambleton, Nicas, and Park
<code>\citeyear{BodeHambNicaPark05}</code>	2005
<code>\citealt{BodeHambNicaPark05}</code>	Boden et al., 2005
<code>\citealp{BodeHambNicaPark05}</code>	Boden et al., 2005

D’autres commandes sont disponibles, voir la documentation du *package*.

Le package natbib : les options

Différentes options permettent de modifier l’aspect des citations :

authoryear, **numbers**, **super** : ces options déterminent le type de citations, “auteurs-année”, numérique, exposant.

round, **square**, **curly**, **angle** : ces options déterminent le matériel avant et après les citations, `(...)`, `[...]`, `{...}`, `<...>`.

semicolon, **comma** : ces options déterminent le séparateur entre clés multiples, `;`, `,`.

sort, **sort&compress**, **compress** : ces options activent la mise en ordre numérique et/ou la compression des clés numériques successives.

sectionbib : redéfinit `\thebibliography` pour produire une `\section*` plutôt qu’un `\chapter*`.

➔ À utiliser avec le *package* **chapterbib** (voir plus loin).

merge : permet de placer dans une même entrée bibliographique plusieurs entrées de la base de données **.bib** en préfixant la clé d’une `*`.

`\citep{Wein67a,*Sa1a68a}` ➔ l’entrée **Sa1a68a** est placée sous l’entrée **Wein67a** dans la liste des entrées bibliographiques.

N’a d’intérêt qu’avec un style du type **unsrnat.bst**.

D’autres options sont disponibles, consulter la documentation.

Le package natbib : changer le style des citations

Les options du *package* permettent de modifier le style des citations.

Ces styles sont surpassés par la commande `\setcitetstyle{-}`.

L’argument est une liste d’éléments du type `option ou clé=valeur` :

authoryear, **numbers**, **super** : pour le type de citations.

round, **square**, **open={carc.}**, **close={carc.}** : correspond au matériel à gauche et à droite de la citation.

semicolon, **comma**, **citesep={carc.}** : matériel entre les citations.

aysep={carc.} : matériel entre les auteurs et l’année.

yysep={carc.} : matériel entre les années d’un auteur commun.

notesep={texte} : matériel entre la référence et la note optionnelle de `\cite`.

Les valeurs par défaut sont :

authoryear, **round**, **comma**, **aysep={;}**, **yysep={,}**, **notesep={, }**.

La commande `\citenumfont` définit la police des citations numériques.

`\renewcommand{\citenumfont}[1]{\textit{#1}}`

Il est possible de nommer des styles en les définissant dans un fichier de configuration et de les installer avec une commande `\citetstyle{-}`.

Le package natbib : personnaliser la bibliographie

Il est possible de modifier l'aspect de l'environnement thebibliography en redéfinissant les commandes suivantes :

- \bibsection permet de modifier le type de la bibliographie : par défaut, c'est \section* ou \chapter* selon la classe du document.
`\renewcommand{\bibsection}{\section{\refname}}`
 ➔ section numérotée.
- La commande \bibpreamble installe du texte entre la commande précédente et la liste bibliographique.
`\renewcommand{\bibpreamble}{%
 {{\bfseries Voici une liste de références à consulter.}}`
- La commande \bibfont définit la police de la bibliographie.
`\renewcommand{\bibfont}{\small\sffamily}`
- La commande \bibnumfmt définit l'aspect des clés numériques.
`\renewcommand{\bibnumfmt}[1]{\textbf{\#1}:}`
- \bibhang est une longueur qui fixe le décalage vers la gauche des entrées de type "auteurs-année".
`\setlength{\bibhang}{2em}`
- \bibsep est une longueur qui fixe l'espace vertical entre les entrées bibliographiques.
`\setlength{\bibsep}{0.5ex}`

Des packages pour gérer les citations

Il existe de nombreux *packages* permettant d'améliorer la gestion de la bibliographie.

⚠ Il peut y avoir des incompatibilités entre ces *packages*.

babelbib rend la bibliographie compatible avec **babel** : il bascule la langue de chaque entrée bibliographique afin de respecter les césures de la langue correspondante.
 Requiert un champ supplémentaire dans les entrées de la base : Language prend les valeurs english, french... dans les options de **babel**.
 ➔ Énorme travail sur la base.

mciteplus permet de regrouper des entrées bibliographiques dans le même \bibitem alors qu'elles sont séparées dans la base .bib (voir aussi **natbib**).
`\cite{Glashow,*Salam,*Weinberg}`
 ➔ Regroupe les références "Salam" et "Weinberg" sous "Glashow".
 ➔ Nécessite des fichiers de style .bst adaptés.

hyperref a une option backref qui permet de lister dans la bibliographie les endroits où les entrées sont citées.
`\usepackage[backref=page]{hyperref}`
 ➔ liste pour chaque entrée les pages où les commandes \cite sont présentes.

notes2bib permet d'ajouter des notes dans la bibliographie avec une commande \bibnote{-}, et d'y envoyer les \footnote aussi.

Plusieurs bibliographies dans un document

Il existe plusieurs *packages* pour créer plusieurs bibliographies dans un seul document.

multibib introduit une commande \newcites{-}{-} qui permet de créer de nouveaux types de bibliographies.

\newcites{A}{Autre bibliographie} définit les commandes :

\citeA, \nociteA, \bibliographystyleA, \bibliographyA

qui clonent les commandes usuelles,

le titre de la bibliographie est le second argument de \newcites.

➔ Il faut lancer bibtex sur le fichier A.aux.

L'option labeled ajoute le "A" ci-dessus aux clés visibles : [A2]...

L'option resetlabels remet les compteurs à 1 pour chaque bibliographie.

splitbib propose de subdiviser les entrées bibliographiques par catégories.

Dans le préambule, l'environnement \begin{category}[-]{-}... \end{category}

permet de regrouper des entrées bibliographiques dans des catégories et d'assigner un titre aux catégories.

Dans le corps du texte, on cite les références comme d'habitude.

L'environnement thebibliography s'occupe alors de découper les entrées par catégories, en ajoutant un préfixe (option de category) aux clés visibles.

Plusieurs bibliographies dans un document (suite)

bibunits permet de créer des "unités" dans le texte dans lesquelles les citations seront regroupées pour former une bibliographie locale.

L'environnement \begin{bibunit}[-]... \end{bibunit} définit une unité : l'argument optionnel est le fichier de style .bst, la commande \putbib{-} correspond dans l'environnement à \bibliography{-}.

➔ Lancer bibtex sur les fichiers bu1.aux, bu2.aux...

➔ Création automatique d'une unité bibliographique par chapitres ou par sections.

➔ Création d'une bibliographie globale avec les commandes \cite*, \nocite*, \bibliography et \bibliographystyle.

chapterbib propose de créer une bibliographie pour chaque fichier inséré par une commande \include{-}.

Dans chacun de ces fichiers, \bibliography et \bibliographystyle produisent une bibliographie.

➔ Lancer bibtex sur chacun des fichiers .aux associés aux fichiers inclus.

Il propose aussi une autre découpe en délimitant la bibliographie locale au sein d'un environnement cbunit.

8.4 Gérer sa base bibliographique

Les problèmes de la gestion d'une base bibliographique

Les bases de données .bib sont des fichiers de texte, comme les fichiers sources \TeX .

➔ N'importe quel éditeur de texte permet donc d'ouvrir et de modifier une telle base.

Cependant, gérer une base à la main est fastidieux et peu efficace :

- Le fichier peut faire plusieurs milliers de lignes et contenir des centaines d'entrées.
➔ Visuellement impossible de retrouver une entrée particulière.
- Lors de la création d'une nouvelle entrée, il faut se souvenir de la syntaxe, des champs obligatoires et des champs optionnels...
- Il y a des risques de doublons, y compris et surtout sur les clés de citations internes, qui *doivent rester uniques*.
- Il y a des risques d'introduire des erreurs syntaxiques qui mettront \bibTeX en échec.
- Certaines références doivent être déplacées vers la fin (\Crossref).
- Il est quasiment impossible de trier, d'extraire, de modifier en bloc, d'exporter dans d'autres formats les entrées bibliographiques...

La gestion manuelle d'une base de données bibliographique est hautement déconseillée.

Des outils pour gérer la base bibliographique

Il existe des logiciels (gratuits) pour gérer de telles bases, entre autres :

- JabRef** ¹⁶ (🍏🍏🍏) est écrit en JAVA ➔ toutes les plateformes.
- BibDesk** ²¹ (🍏) a déjà été mentionné.

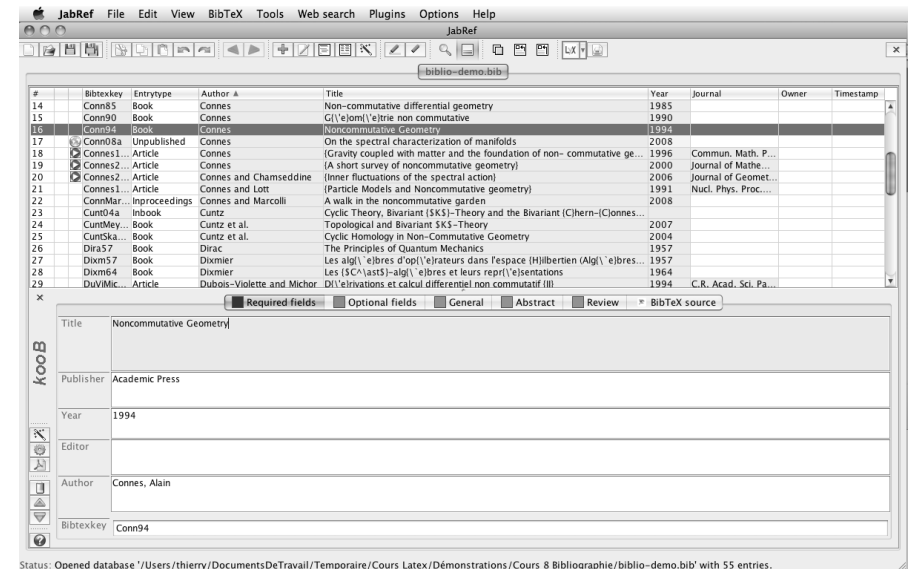
Ces logiciels ont les fonctionnalités suivantes :

- Présentation en liste des entrées bibliographiques.
➔ Recherche visuelle, tri selon différents critères (selon les colonnes).
- Lors de la création d'une nouvelle entrée, les champs obligatoires sont mis en évidence et le logiciel vérifie qu'ils ne sont pas vides.
- Génération automatique des clés de citations internes selon un schéma choisi.
- Changement global de champs textuels : noms d'auteurs, noms des journaux...
- Extraction de citations à partir d'un fichier .aux.
➔ Base spécifique à un projet pour échanges avec d'autres auteurs.
- Interface avec le WEB : import en ligne (**ScienceDirect** ¹³⁹, **SpringerLink** ¹⁴⁰, **MathSciNet** ¹⁴¹, **Spirex** ¹⁴², **arXiv** ⁹⁰...), champs url d'une entrée...
- Exportation dans différents formats : texte brut, .rtf et .doc (**OpenOffice** ⁴⁶, **Microsoft Word** ⁴⁵), HTML (site WEB), autres formats de base de données bibliographiques (**RIS** ¹⁴³, **EndNote** ¹⁴⁴)...

Aujourd'hui des services en lignes proposent la gestion d'une base bibliographique : **Mendeley** ¹⁴⁵, **CiteULike** ¹⁴⁶, **Zotero** ¹⁴⁷ (extension **FireFox** ¹⁴⁸)...

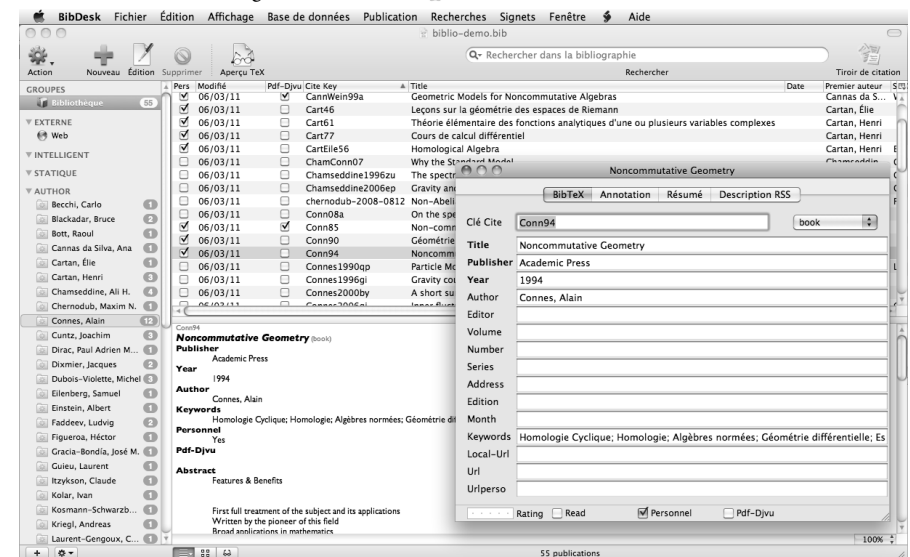
Le logiciel JabRef

L'interface de travail du logiciel JabRef ¹⁶ (🍏🍏🍏) :



Le logiciel BibDesk

L'interface de travail du logiciel BibDesk ²¹ (🍏) :



8.5 L'avenir avec biblatex

Les problèmes avec bibtex

On a vu les avantages d'une approche de la bibliographie fondée sur une base de données.

Mais l'outil `bibtex` a de gros défauts :

- À des corrections de bugs près, la dernière version ¹⁴⁹ de `bibtex` date de 1988 et porte actuellement le numéro 0.99d !
→ C'est un projet qui n'aboutira probablement jamais.
- Ce logiciel est en 7 bits et son remplaçant `bibtex8` qui fonctionne sur 8 bits ne gère pas l'UNICODE...
→ `bibtexu` a vu le jour pour le support UNICODE, mais semble abandonné...
- Le langage de script présent dans les fichiers de style `.bst` est compliqué et très éloigné de la culture L^AT_EX.
→ Difficile de créer soi-même un tel fichier de style.
→ Au mieux, sans investissement lourd, on peut en adapter des petits morceaux.
- Pas de support natif pour le multilinguisme.
- Lorsqu'on envoie un fichier `.bbl` avec le document source `.tex`, l'intérêt de la base de données est perdu : la bibliographie est déjà formatée avec un style déterminé.

Le couple biblatex et biber

En 2006 a démarré un nouveau projet de bibliographie sous L^AT_EX qui propose du neuf à partir de l'ancien.

- On conserve l'acquis principal d'une bibliographie reposant sur une base de données.
→ Le format de la base est `.bib`, auquel sont apportés des évolutions (nouveaux types, nouveaux champs...).
- On délègue à L^AT_EX le soin de mettre en forme les entrées bibliographiques.
C'est le rôle du *package* **biblatex**.
→ Disparition des fichiers de style `.bst`.
→ Syntaxe plus habituelle pour modifier le style, sous forme de commandes L^AT_EX.
→ Interaction avec les *packages* habituels : **babel**, **hyperref**...
- Un nouveau moteur de traitement de la base, appelé `biber`, est créé pour faire l'interface entre le fichier `.bib` et le fichier `.bbl`.
- Gestion native de l'UNICODE.
- Le *package* **biblatex** reprend et uniformise les fonctionnalités de nombreux *packages* améliorant la gestion des citations et les rend inutiles pour la plupart : **cite**, **natbib**, **babelbib**, **mciteplus**, **multibib**, **splitbib**...

Le style de la bibliographie est déterminé par un fichier `.bbx`.

Le style des citations est déterminé par un fichier `.cbx`.

→ Ce sont des fichiers écrits avec des commandes L^AT_EX...

Le fonctionnement de biblatex

Le principe premier est le suivant :

Les entrées de la base bibliographiques utilisées dans le document sont *intégralement* insérées dans le fichier `.bbl` dans un format exploitable par L^AT_EX.

Le fichier `.bbl` est construit à partir d'une ou plusieurs bases `.bib` :

- soit avec le nouveau moteur `biber`, non encore finalisé ;
- soit avec l'ancien moteur `bibtex8` et un fichier de style inséré par **biblatex**.

`biber` vient d'apparaître (mars 2011) dans l'installation **TeXLive** ¹¹ grâce à `tlmgr`.

Le fichier `.bbl` ne dépend (presque) pas du style choisi → transportable sans perte.

Dans le fichier source `.tex`, on n'a donc plus qu'à :

- Appeler le *package* **biblatex** :

```
\usepackage[-]{biblatex}
```

→ Les options déterminent le comportement de la bibliographie et des citations.

- Désigner *dans le préambule* les fichiers `.bib` à insérer :

```
\bibliography{fichierun,fichierdeux,...}
```

- Insérer la bibliographie à l'endroit voulu avec :

```
\printbibliography[-]
```

→ Les options déterminent les entrées bibliographiques listées, leur ordre, leur style et l'aspect de la bibliographie (ou des bibliographies).

Une entrée dans .bbl avec biblatex

biblatex produit ce type d'entrées dans un fichier `.bbl` :

```
\entry{Cart51b}{inproceedings}{}
  \name{author}{1}{%
    {{Cartan}{C.}%
    {Henri}{H.}%
    }{}%
    {}{}%
  }
  \list{publisher}{1}{%
    {Masson}%
  }
  \strng{namehash}{CH1}
  \strng{fullhash}{CH1}
  \field{sortinit}{C}
  \field{booktitle}{Colloque de Topologie (Bruxelles 1950)}
  \field{title}{La transgression dans un groupe de {L}ie et
    dans un espace fibré{'e} principal}
  \field{year}{1951}
\endentry
```

Les ajouts au format .bib : des nouveaux champs

biblatex enrichit la base bibliographique avec de nouveaux champs, parmi lesquels :

`eprint`, `eprinttype` : identification d'une publication en ligne.

→ Peut être utilisé pour les articles sur **arXiv**⁹⁰ :

`eprint = {math/0307200v3},`

`eprinttype = {arxiv},`

`eprintclass` peut compléter les informations si nécessaire.

`url` : ce champs existait déjà comme option, et il n'était pas toujours reconnu par les fichiers de style .bst.

Il fait désormais partie intégrante des champs présentés dans les références.

`language` : la langue de la référence.

`hyphenation` : définit la langue vers laquelle **babel** doit basculer pour écrire l'entrée bibliographique.

→ La valeur est une langue connue de **babel**.

`edition`, `subtitle`, `authortype`, `editortype`, `doi`... : la liste est très longue !

De nombreux champs nouveaux deviennent des options dans les types usuels.

Les ajouts au format .bib : des nouveaux types

biblatex enrichit la base bibliographique avec de nouveaux types, parmi lesquels :

@online : ressource en ligne (alias : `@electronic`, `@www`).

obligatoire : `author` ou `editor`, `title`, `year` ou `date`, `url`.

optionnel : `subtitle`, `titleaddon`, `language`, `note`.

@set : ce type regroupe sous une clé commune différentes entrées bibliographiques qui seront placées ensemble dans la liste des références.

obligatoire : `entryset`, `crossref`.

optionnel : tous les autres !

`\entryset={cle1,cle2,...}`

@report : rapport dont la nature est spécifiée par le champ `type`.

obligatoire : `author`, `title`, `type`, `institution`, `year` ou `date`.

optionnel : `subtitle`, `titleaddon`, `language`, `number`, `version`, `note`, `location`, `month`, `isrn`, `chapter`, `pages`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprinttype`, `url`, `urldate`.

@thesis : une thèse dont la nature est spécifiée par le champ `type`.

obligatoire : `author`, `title`, `type`, `institution`, `year` ou `date`.

optionnel : `subtitle`, `titleaddon`, `language`, `note`, `location`, `month`, `chapter`, `pages`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprinttype`, `url`, `urldate`.

Les options principales de biblatex

Le **package biblatex** regorge d'options, sous forme de `clé=valeur` :

`style` : détermine l'aspect de la bibliographie en appelant des fichiers .bbx et .cbx.

`bibstyle` : désigne directement le fichier .bbx.

`citestyle` : désigne directement le fichier .cbx.

`backend` : détermine le moteur bibliographique (`biber`, `bibtex8`, `bibtex`).

`sorting` : détermine la façon de classer les entrées bibliographiques.

→ "nom-titre-année", "nom-année-titre", "nom-année-volume-titre"...

`sortcites=true` : réordonne les citations numériques multiples.

`hyperref`, `backref`, `backrefstyle` : permet d'activer les hyperliens externes (champs `url`) et internes, de lister les pages où les références sont citées...

`arxiv` : détermine vers quelle page sur **arXiv**⁹⁰ (`abs`, `ps`, `pdf`) le champ `arxiv` renvoie.

`natbib=true` : installe les commandes de citations de **natbib** : `\citet`, `\citep`, `\citealt`, `\citealp`...

`minnames`, `maxnames` : si la liste des auteurs dépasse `maxnames`, on n'en affiche que `minnames` avec un "*et al.*" derrière.

`language`, `babel` : ces options contrôlent le comportement des entrées bibliographiques vis à vis du multilinguisme.

`isbn=true`, `url=true`, `doi=true` : déterminent si on imprime ou non les champs correspondants.

biblatex contient de nombreuses autres options !

Les styles usuels

Quelques styles bibliographiques définis par **biblatex** (`style=...`) :

numeric : c'est le style de citations usuel basé sur les clés numériques.

Il définit une option `subentry=true`, `false` : si on cite un élément d'une entrée à l'intérieur d'un `@set`, il affiche [5c] (`true`) ou [5] (`false`).

numeric-comp : version compacte de **numeric** où les citations multiples sont présentées par intervalles. [8,3,1,7,2] → [1-3,7,8].

numeric-verb : version "étendue" de **numeric** où les citations multiples sont présentées séparées. [2,5,6] → [2]; [5]; [6].

alphabetic : style semblable à celui de `alpha.bst`. Les clés sont des versions compactes de lettres et de chiffres obtenus à partir des auteurs et de l'année.

alphabetic-verb : version "étendue" de **alphabetic**.

authoryear : c'est le style "auteurs-année".

authoryear-comp : version compacte de **authoryear**.

"Doe 1995b; Doe 1992; Jones 1998; Doe 1995a" → "Doe 1992, 1995a,b; Jones 1998".

verbose : se comporte comme **authoryear** dans la liste des références, et cite toute la référence dans le texte la première fois, puis une version abrégée les autres fois.

De nombreux styles sont fournis pour les revues usuelles.

Les commandes de citation

bibl_{at}ex définit de nouvelles commandes de citation.

`\cite[pré][post]{-}` : imprime une citation, sans parenthèse dans le mode “auteurs-année”. *pré* et *post* sont des textes placés avant et après la citation.

`\cite[p. 100]{Bert96}` ➔ [Ber96, p. 100] (syntaxe ordinaire).

`\parencite[pré][post]{-}` : imprime une citation entre parenthèses dans le mode “auteurs-année”. *pré* et *post* sont placés dans les parenthèses.

`\footcite[pré][post]{-}` : imprime la citation dans une note de bas de page.

`\textcite[pré][post]{-}` : imprime, selon le style choisi, le nom des auteurs ou des éditeurs, suivi entre parenthèses du numéro de la référence ou de l'année.

`\supercite{-}` : imprime, dans le style numérique, le numéro en exposant.

`\cites(pré)(post)[pré][post]{-}...[pré][post]{-}...` : toutes les commandes précédentes avec un *s* à la fin permettent de citer une série d'entrées bibliographiques.

`\cite*`, `\parencite*` : n'impriment que l'année dans le style “auteurs-année”.

`\citeauthor`, `\citetitle`, `\citeyear`, `\citeur` : impriment la partie correspondante de l'entrée bibliographique.

`\fullcite`, `\footfullcite` : impriment toute l'entrée bibliographique dans le texte ou dans une note de bas de page.

Découper une bibliographie par sections

Le but est de créer une bibliographie indépendante par bloc sémantique.

Un bloc sémantique peut être une section ou un chapitre...

➔ Regrouper des articles dans un compte-rendu de conférence.

Les bibliographies sont incluses dans les blocs (ici des chapitres) :

```
\chapter{Chapitre un}
\begin{refsection}
...
\printbibliography[heading=subbibliography]
\end{refsection}
...
```

Les bibliographies sont imprimées à la fin du document :

```
\chapter{Chapitre un}
\begin{refsection}
...
\end{refsection}
...
\printbibliography[section=1]
...
```

Découper une bibliographie par segments

Le but est de créer une bibliographie globale découpée par segments du texte.

Un segment est défini par l'environnement `refsegment`.

➔ Créer à la fin d'un livre une bibliographie par chapitres.

```
\chapter{Chapitre un}
\begin{refsegment}
...
\end{refsegment}

\chapter{Chapitre deux}
\begin{refsegment}
...
\end{refsegment}
...
\printbibliography[segment=1]
\printbibliography[segment=2]
...
```

Découper une bibliographie par types

Il est possible de subdiviser une bibliographie selon différents critères :

Type de référence : il s'agit de regrouper les entrées bibliographiques par leur type, `@article`, `@book`, `@online`...

```
\printbibliography[type=article]
\printbibliography[type=book]
\printbibliography[nottype=article,nottype=book]
```

Mots-clés : il s'agit de regrouper les entrées bibliographiques ayant (ou non) certains mots-clés dans le champ `keywords`.

```
\printbibliography[keyword=homologie]
\printbibliography[notkeyword=homologie]
```

Regroupement dans le document : il s'agit de regrouper les entrées bibliographiques en catégories définies dans le document.

```
\DeclareBibliographyCategory{mathematique}
\addtocategory{mathematique}{cle1,cle2,...}
...
\printbibliography[category=mathematique]
\printbibliography[notcategory=mathematique]
```

On peut mélanger tous ces critères...

.....
Autres fonctionnalités

En plus d'utiliser les styles prédéfinis, il est possible de modifier l'aspect des citations et de la bibliographie à l'aide de diverses commandes :

`\bibsetup` : code exécuté au début de la bibliographie pour la mettre en forme.
`\citetsetup` : code exécuté avant chaque citation pour la mettre en forme.
`\bibfont` : code pour changer la typographie de la bibliographie.
`\bibleftparen`, `\bibrightparen` : parenthèses ouvrantes et fermantes.
`\biblabelsep` : dimension horizontale entre les entrées et leur clé visible.
`\bibitemsep` : dimension verticale entre les entrées bibliographiques.

Autres commandes : de très nombreux éléments de ponctuations, des distances insérées à divers endroits dans la composition des entrées bibliographiques et des citations...

Il est possible d'imprimer une bibliographie avec tous les champs.
Il est possible d'ajouter des préfixes aux clés visibles.
Il est possible de définir et de nommer des filtres de sélection d'entrées bibliographiques.
Il est possible de générer un index automatique à partir des entrées bibliographiques.

Nombreux exemples concernant les styles et l'utilisation de **biblatex** dans le dossier : `texmf-dist/doc/latex/biblatex/examples/`.
➔ Il est recommandé de consulter la documentation et ces exemples pour profiter pleinement de toutes les fonctionnalités et de toute la richesse de **biblatex**.

Cours 9 – La classe beamer

"If your words or images are not on point,
making them dance in color won't make them relevant.
Audience boredom is usually a content failure,
not a decoration failure."
Edward R. Tufte
PowerPoint Is Evil (www.wired.com)

9.1 Un tour d'horizon des possibles	180
Où l'on pèse d'abord les avantages et les inconvénients de préparer un exposé avec son ordinateur, et où l'on se demande alors quel logiciel utiliser...	
9.2 Les commandes de base	183
Où l'on apprend à réaliser sa présentation avec quelques principes de base et à l'aide de quelques commandes nouvelles...	
9.3 Modifier l'aspect de sa présentation	195
Où l'on apprend à modifier l'apparence locale ou globale de sa présentation en faisant bon usage des modèles et des thèmes...	
9.4 Au delà de la présentation	201
Où l'on découvre qu'il est aussi possible de générer des notes écrites avec le texte de la présentation ou bien que l'on peut composer un poster à l'aide de diapositives...	

9.1 Un tour d'horizon des possibles

Une question fondamentale

Pourquoi utiliser un ordinateur pour une présentation ?

- Parce qu'on n'a pas le choix : absence de tableau, absence de rétroprojecteur.
- Parce qu'on souhaite recycler :
 - soit la présentation elle-même, sous différentes formes modifiées ;
 - soit des morceaux d'une publication (formules, dessins, tableaux...).
- ➔ Requiert des logiciels compatibles pour les articles et les présentations...
- Parce qu'on a du matériel spécifique : formules compliquées, dessins, vidéos...
- Parce qu'on souhaite une présentation dynamique au service d'une pédagogie pensée.
- Pour lutter contre l'entropie qui place toujours les notes manuscrites et les transparents dans un désordre maximal.
- Parce qu'il est plus facile d'accélérer un exposé trop long en appuyant frénétiquement sur la touche "suivant" qu'en jonglant avec des transparents.
- Parce que c'est très reposant de n'avoir qu'à appuyer sur une touche...
- Parce qu'on est allergique à la craie.
- Parce qu'on a une écriture de cochon.
- Parce qu'on est tendance...

⚠ **PowerPoint is Evil**¹⁵⁰ : une critique des présentations à l'ordinateur...

Les logiciels de présentation

Principe des logiciels de présentation :

- Le document est découpé en diapositives (*slides*).
- Les diapositives sont projetées en mode plein écran grâce à un vidéo projecteur.

Quelques logiciels courants :

PowerPoint¹⁵¹ : (🍏) Il fait partie de la suite bureautique **Office**¹⁵² de Microsoft.

Quelques problèmes de compatibilités entre les versions WINDOWS et MAC.

Impress¹⁵³ : (🍏) C'est le module de présentation de **OpenOffice**.

Il est plus ou moins compatible avec le format de **PowerPoint**.

Keynote¹⁵⁴ : (🍏) Il fait partie de la suite bureautique **iWork**¹⁵⁵ d'Apple.

N'existe pas sur d'autres systèmes d'exploitation.

Leurs avantages évidents : Mise en page aisée, animations diverses à l'intérieur des diapositives et entre les diapositives, insertion possible de vidéos, choix d'un thème général qui modifie l'aspect global de la présentation, export en PDF...

Leurs défauts principaux : Logiciel compatible requis sur l'ordinateur de la présentation.

Difficile d'insérer des formules de mathématique, surtout dans le flux de texte.

- ➔ Les formules LaTeX peuvent être insérées en tant qu'images.
 - ➔ Utiliser les mêmes polices que celles de LaTeX : **LATIN MODERN**¹⁵⁶ en **OPENTYPE**⁵⁹.
- Ne semblent pas fournir une table des matières avec hyperliens internes.

La classe beamer

La classe **beamer**¹⁵⁷ répond à un besoin très ancien de créer une présentation à l'ordinateur à partir d'un document écrit en LaTeX .

Quelques ancêtres de **beamer** : la classe **slides** (1997), la classe **prosper** (2003), le *package* **texpower** (2005), Foil TeX et sa classe **foils** (2008)...

➔ **beamer** les dépasse tous en fonctionnalités.

➔ **beamer** propose un mode de compatibilité avec ces anciennes approches.

Les principes généraux de **beamer** sont les suivants :

- Le document produit est du PDF, qui est projeté avec n'importe quel logiciel¹⁵⁸ capable d'afficher un PDF en plein écran.
 - ➔ Aucune dépendance à un logiciel particulier, portabilité du document.
- La classe **beamer** ajoute des commandes spécifiques, mais l'essentiel reste du LaTeX tout à fait standard.
 - ➔ Apprentissage basique très rapide.
 - ➔ Possible de copier-coller directement à partir d'un autre document LaTeX .
- La classe **article**, associée au *package* **beamerarticle**, produit un document à imprimer reprenant le texte de la présentation.
 - ➔ Possibilité de distribuer des notes imprimées plus fournies en texte.
- Notion de thèmes, modifiables à volonté avec le langage LaTeX .

Les particularités de beamer

Une présentation à l'ordinateur admet certaines contraintes :

- Une présentation à l'ordinateur se fait en mode "paysage".
 - Les projecteurs ont en général une résolution de 1024×768 pixels (rapport 4×3).
 - ➔ **beamer** fixe la taille de la page à $128\text{mm} \times 96\text{mm}$.
- Il est préférable d'utiliser des fontes sans empattements, y compris pour les mathématiques. ➔ **beamer** se charge de basculer vers ces fontes.

⚠ éviter $\backslash\mathrm{-}$ ➔ $\backslash\text{-}$, $\backslash\text{DeclareMathOperator}\{-\}\{-\}$.
- La taille des caractères doit être assez grande.
 - ➔ **beamer** crée un document en 11pt de taille "physique" réduite : à la projection, l'ensemble est agrandi et le texte est très lisible.
 - Pour un projecteur de 1024×768 pixels, les caractères font 31 pixels de haut.
 - ⚠ À l'impression, penser à agrandir aussi les pages !
- Le texte est collé à gauche et non pas justifié.
- Il est ennuyeux d'avoir à appuyer de façon répétitive sur une touche (flèches) pour passer d'une diapositive à une autre afin de répondre à une question.
 - beamer** est capable de produire automatiquement une table des matières et de fournir une barre de navigation avec des hyperliens internes.
 - ➔ Navigation très aisée au sein du document.

Quelques problèmes de beamer

Du fait de sa dépendance au format de fichier PDF, la classe **beamer** souffre des quelques problèmes suivants :

Les effets de transition : Le format PDF n’est pas conçu pour disposer d’effets de transition à l’intérieur des diapositives ou entre les diapositives.

Adobe Reader ¹⁵⁹ accepte quelques effets entre diapositives.

⚠ Ces effets ne fonctionnent pas nécessairement dans d’autres logiciels.

➔ **beamer** permet de les utiliser.

➔ Certains logiciels permettent d’ajouter des effets “après coup” à la main.

Le multimédia : Les afficheurs PDF n’ont pas tous des fonctionnalités multimédia.

Adobe Reader est capable de présenter certaines vidéos incluses dans le document sur WINDOWS et MAC, mais pas sur LINUX.

➔ Il est possible de déléguer ce travail à un logiciel externe.

Des polices parfois absentes : Parfois certains caractères d’écriture ne s’affichent pas de la même façon d’un ordinateur à un autre (symboles ZAPF DINGBATS par exemple).

Le document PDF n’inclut pas nécessairement toutes les fontes utilisées.

Les moteurs **pdf_latex** et **dvipdfm** insèrent toutes les fontes, mais pas **dvips**.

L’option **dvips -Pdownload35** permet d’insérer toutes les fontes.

Voir aussi le fichier de configuration **updmap.cfg**.

9.2 Les commandes de base

La structure d’une présentation

Une présentation sous **beamer** peut être découpée globalement avec des commandes **\part**, **\section** et **\subsection** (**\subsubsection** existe mais l’éviter).

➔ **\part** n’est utile que dans des séries d’exposés (cours par exemple).

L’unité élémentaire d’une présentation est la diapositive, définie par la commande **\frame[-]{-}** ou l’environnement **\begin{frame}[-]{-} ... \end{frame}**.

➔ Dans la suite on ne considère que l’environnement **frame**.

Les éléments suivants peuvent être générés automatiquement :

Une diapositive de titre : il faut renseigner le titre, l’auteur, l’institut, la date...

Des tables des matières : il est possible de créer des diapositives contenant des tables de matières de différents niveaux et aspects.

Des diapositives “section” : elles peuvent marquer l’entrée dans une section.

Une annexe : une annexe peut être créée pour “cacher” des diapositives placées au delà de la fin de l’exposé (réserve pour les questions).

➔ Absente des tables des matières, non comptabilisée dans le total des diapositives.

Concept de “pauses” : chaque diapositive peut être subdivisée pour faire apparaître du texte étape par étape (1 étape = 1 page PDF).

Le compteur **beamerpauses** indexe ces étapes : il augmente d’une unité par étape définie.

Un code source type

```
\documentclass[10pt]{beamer}
\usepackage[utf8]{inputenc}
\usepackage[T1,T1]{fontenc}
\usepackage[english,french]{babel}
\usepackage{lmodern}

\usetheme{Goettingen}

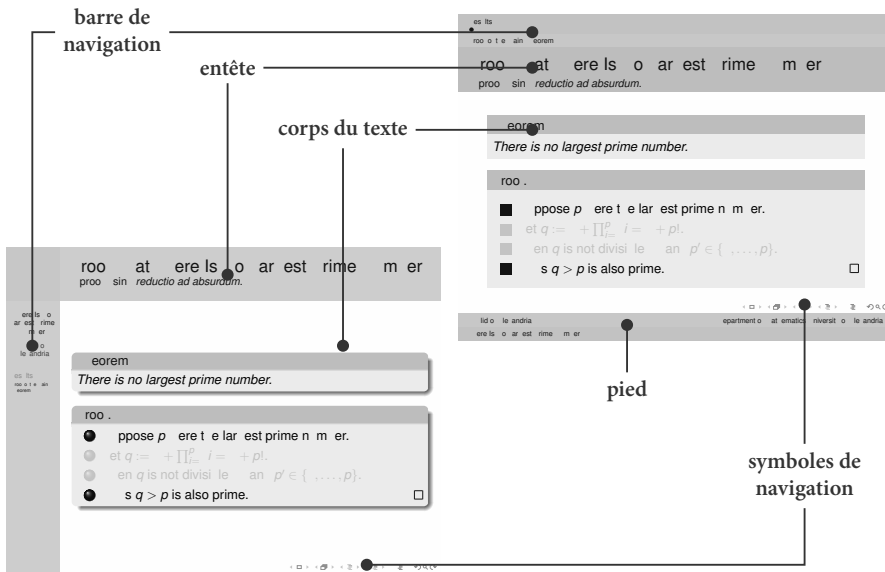
\title[Beamer en action]{Beamer en action}
\author[T. Masson]{Thierry Masson}
\institute[CPT]{Centre de Physique Théorique}
\date[mai 2011]{mai 2011}

\begin{document}
\section{Une première section}
\begin{frame}{Une première diapositive}
  Bonjour !
\end{frame}
\end{document}
```

beamer charge les *packages* suivants :

- geometry**
- graphicx**
- xcolor**
- pgf**
- hyperref**
- amssymb**
- amsmath**
- amsfonts**
- amsthm**
- translator**
- enumerate**

La structure d'une diapositive en images



La structure d'une diapositive en théorie

Une diapositive est constituée de différents éléments "physiques" :

La barre de navigation : cette barre contient en général le rappel de la structure logique de la présentation : sections, sous sections, diapositives.

Cette barre peut être horizontale ou verticale selon le thème choisi.

L'entête : c'est l'endroit usuel pour placer le titre (et le sous titre) de la diapositive.

Le corps du texte : c'est le contenant de ce qui est présenté dans la diapositive.

Le pied : on peut y placer le titre de la conférence, le nom du présentateur et son institut, la date, le numéro de la diapositive...

Les symboles de navigation : ce sont des symboles (discrets) cliquables permettant la navigation dans la présentation : début, section, table des matières...

Ces éléments ont une apparence et une position déterminées par le thème choisi.

Ils peuvent être personnalisés grâce à des modèles (*templates*) facile à modifier.

De nombreux textes (titre, sections, sous sections...) et symboles insérés dans ces éléments sont des hyperliens internes au document.

➔ Accès facile et permanent à diverses parties du document.

L'environnement frame

Plusieurs syntaxes sont possibles pour l'environnement frame :

```
\begin{frame}{Titre}
...
\end{frame}
```

```
\begin{frame}{Titre}{Sous Titre}
...
\end{frame}
```

```
\begin{frame}
\frametitle{Titre}
\framesubtitle{Sous Titre}
...
\end{frame}
```

Les commandes `\frametitle` et `\framesubtitle` sont optionnelles.

L'option `[plain]` ôte les barres de navigation et le pied de la diapositive.

```
\begin{frame}[plain]{Titre}
...
\end{frame}
```

```
\begin{frame}[plain]
...
\end{frame}
```

➔ Gain d'espace pour le contenu de la diapositive (présentation d'un graphique par ex.).

Par défaut, le texte est centré verticalement dans une diapositive.

Les options `t`, `b` et `c` (défaut) de `frame` peuvent modifier ce comportement.

La diapositive de titre

La diapositive de titre peut-être créée par :

```
\begin{frame}[plain]
\titlepage
\end{frame}
```

On peut écrire plus simplement :

```
\maketitle
```

Les champs à renseigner pour créer la diapositive de titre sont :

`\title[-]{-}` définit le titre de la présentation.

`\subtitle[-]{-}` définit le sous titre de la présentation.

`\author[-]{-}` désigne la liste des auteurs séparés par `\and`.

`\institute[-]{-}` désigne la liste des instituts séparés par `\and`.

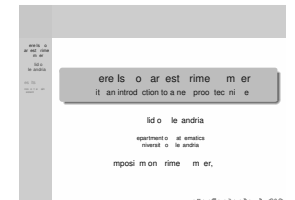
`\date[-]{-}` définit la date, et on ajoute souvent le lieu de la conférence.

`\titlegraphic{-}` désigne un graphique de titre.

Les arguments optionnels dans les commandes ci-dessus sont utilisés pour le rappel des données sur chaque diapositive. ➔ Fournir une version plus courte.

La composition de la diapositive de titre dépend du thème choisi.

Il existe des commandes `\partpage`, `\sectionpage` et `\subsectionpage`.



Les tables des matières

Les tables des matières sont composées des titres des sections et des sous-sections.

`\tableofcontents[options]` avec les options :

`sectionstyle=` détermine le style de la section courante et des autres sections :

sect. courante/autres sect. avec les valeurs `show`, `shaded`, `hide`.

Si le second est omis, il prend la valeur du premier.

`subsectionstyle=` détermine de même l'aspect des sous sections :

ss-sect. courante/autres ss-sect. dans la sect. courante/autres ss-sect..

Si le troisième est omis, il prend la valeur du second.

Si le second est omis aussi, les trois prennent la valeur du premier.

`currentsection` équivaut à

`sectionstyle=show/shaded,subsectionstyle=show/show/shaded.`

`currentsubsection` équivaut à `subsectionstyle=show/shaded.`

`hideallsubsections` équivaut à `subsectionstyle=hide.`

`hideothersubsections` équivaut à `subsectionstyle=show/show/hide.`

Avec ces options, les tables des matières n'affichent pas nécessairement la même chose à différents endroits de la présentation.

D'autres options sont disponibles.

Les tables des matières ont des hyperliens internes au document.

Les colonnes

beamer propose une mise en page (locale) sur plusieurs colonnes.

```
\begin{columns}[t]
  \begin{column}[c]{0.6\textwidth}
    ...
  \end{column}
  \begin{column}{0.35\textwidth}
    ...
  \end{column}
\end{columns}
```

L'option de l'environnement `columns` définit la position horizontale des colonnes qu'il contient : `t`, `b`, `c` (défaut), `T` (alignement haut première ligne).

`columns` supprime les marges gauche et droite ➔ on dispose de toute la page en largeur.

L'option `onlytextwidth` réduit à la zone de texte.

L'option de l'environnement `column` définit la position horizontale (mêmes options).

On peut placer autant de `column` qu'on veut (peut) dans `columns`.

Les `column` contiennent n'importe quel matériel L^AT_EX : texte, tableau, graphique...

On peut placer plusieurs `columns` (l'un en dessous de l'autre) dans la même diapositive.

➔ À utiliser sans modération pour des vis-à-vis "texte"/"graphique".

Avancer par étapes

Certaines commandes usuelles et certains environnements courants de L^AT_EX ont été étendus pour accepter un nouvel argument optionnel `<>` :

`\cmd<étapes>([-] { - })` où les autres arguments conservent leur place et nombre.

`<étapes>` définit le comportement de `\cmd` par rapport aux étapes de la diapositive.

➔ Fait référence au compteur `beamerpauses` :

`<m-n>` active la commande entre les valeurs *m* et *n* du compteur `beamerpauses`.

`<m->` active la commande à partir de la valeur *m* du compteur `beamerpauses`.

`<-n>` active la commande jusqu'à la valeur *n* du compteur `beamerpauses`.

`<.->` active la commande à partir de la valeur courante du compteur `beamerpauses`.

`<+->` active la commande à partir de la valeur courante du compteur `beamerpauses` et incrémente `beamerpauses` d'une unité. ➔ Syntaxe la plus utile !

`<étapes1,étapes2,...>` combine plusieurs intervalles.

Le compteur `beamerpauses` est défini à 1 au début de l'environnement `frame` et atteint la valeur maximale requise par les commandes `\cmd<étapes>([-] { - })` rencontrées.

La diapositive sera répartie sur autant de pages physiques PDF que la valeur maximale atteinte par ce compteur.

⚠ La syntaxe `<étapes>` est en fait plus générale, et accepte une spécification du mode : `presentation`, `beamer`, `trans`, `handout` et `article`.

Une commande `\pause` permet de gérer les étapes d'une diapositive ➔ non évoquée ici.

Quelques commandes gérant les étapes

`\textbf`, `\textit`, `\textsl`, `\textrm`, `\textsf` acceptent `<étapes>` comme premier argument : `\textbf<2-4>{En gras de 2 à 4}`

`\color<étapes>[modèle]{spécifications}` active la couleur définie par le modèle et les spécifications sur les étapes spécifiées par `<étapes>`.

`\label<étapes>{label}` insère le label *label* sur la page correspondant à l'étape spécifiée par `<étapes>`.

⚠ *étapes* ne doit contenir qu'une seule étape !

`\uncover<étapes>{texte}` affiche *texte* sur les étapes spécifiées par *étapes*.

En dehors de ces étapes, la place pour *texte* est réservée.

Il est possible d'afficher en "transparent" le texte caché (`\setbeamercovered{ - }`).

`\only<étapes>{texte}` affiche *texte* sur les étapes spécifiées par *étapes*.

En dehors de ces étapes, la place pour *texte* n'est pas réservée.

`\temporal<étapes>{texte avant}{texte pendant}{texte après}` affiche *texte avant* avant les étapes spécifiées par *étapes*, *texte pendant* sur les étapes spécifiées par *étapes* et *texte après* après la dernière étape spécifiée par *étapes*.

D'autres commandes existent pour dévoiler et cacher du texte.

Les arguments des commandes `\uncover`, `\only` et `\temporal` peuvent contenir du texte, des tableaux, des images et même des définitions de commandes...

Ces commandes peuvent s'emboîter.

Les listes

beamer retient les trois sortes de liste de L^AT_EX : `itemize`, `enumerate` et `description`.

L'apparence de ces listes et de leurs items est modifiable.

Les items des listes acceptent l'argument `<étapes>` :

<code>\begin{itemize}</code> <code>\item<1-> Un</code> <code>\item<2-> Deux</code> <code>\end{itemize}</code>	<code>\begin{enumerate}</code> <code>\item<+> Un</code> <code>\item<+> Deux</code> <code>\end{enumerate}</code>	<code>\begin{description}</code> <code>\item<1-3> [Un]</code> <code>\item<2-4> [Deux]</code> <code>\end{description}</code>
--	--	--

Autre syntaxe :

<code>\begin{itemize}[<+>]</code> <code>\item Un</code> <code>\item Deux</code> <code>\item Trois</code> <code>\end{itemize}</code>	<code>\begin{enumerate}[<+>]</code> <code>\item Un</code> <code>\item<1-> Deux</code> <code>\item Trois</code> <code>\end{enumerate}</code>
---	---

➔ Raccourci qui fait apparaître les items les uns après les autres sauf mention contraire.

L'énumération reprend les options du *package* **enumerate** (déjà chargé) :

```
\begin{enumerate}[<+>][<i>] ... \end{enumerate}
```

Mettre en valeur du texte

Les thèmes installent une couleur dominante, appelée "couleur de structure".

`\structure<>{-}` met du texte dans la couleur de structure.

`\structure{Couleur structure}` ➔ Couleur structure

`\alert<>{-}` met en valeur du texte en le colorant en rouge par défaut.

`\alert{En alerte !}` ➔ En alerte !

beamer propose de mettre en valeur des blocs de texte de plusieurs lignes :

<code>\begin{block}{Un titre}</code> Du texte en boîte <code>\end{block}</code>	<code>\begin{alertblock}{Un titre}</code> Du texte en alerte <code>\end{alertblock}</code>
---	--

```
\begin{exampleblock}{Un titre}
  Du texte en exemple
\end{exampleblock}
```

Ces trois types de blocs ont chacun une couleur assignée (modifiable).

⚠ Le terme "bloc" est excessif : dans certains thèmes il n'y a pas de bloc dessiné...

Ces trois environnements acceptent un argument `<>` :

```
\begin{block}<>{-} ... \end{block}
```

➔ N'affiche le bloc que selon les spécifications de `<>`, sa place est réservée.

Les espaces réservés

Ajouter ou ôter du texte produit une recomposition de la diapositive ➔ non souhaitable.

```
\begin{overlayarea}{<dimh>}{<dimv>} ... \end{overlayarea}
```

réserve une zone de travail de largeur `<dimh>` et de hauteur `<dimv>`.

➔ Dans cette zone, des `\only` ne changeront pas la mise en page globale.

```
\begin{overlayarea}{\textwidth}{3cm}
  \only<1> {Lorem ipsum dolor sit amet.}
  \only<2-> {Fusce pretium ullamcorper neque sit amet luctus.}
\end{overlayarea}
```

```
\begin{overprint}[-] ... \end{overprint}
```

réserve un espace vertical de hauteur minimale (calculée) adaptée à son contenu.

La largeur vaut `\textwidth` sauf mention contraire dans l'argument optionnel.

Les étapes du contenu sont définies par la commande `\onslide<>` qui se comporte un peu comme `\item<>`. ➔ Les `\onslide<étapes>` doivent être disjoints.

```
\begin{overprint}
  \onslide<1> Lorem ipsum dolor sit amet.
  \onslide<2-> Fusce pretium ullamcorper neque sit amet luctus.
\end{overprint}
```

Les mathématiques

Les mathématiques dans **beamer** sont les mêmes que dans L^AT_EX.

Les *packages* **amsmath** et **amsthm** sont chargés par défaut (sauf avec l'option `noamsthm`).

➔ On peut utiliser **amssymb**, **mathtools**, **xy**...

beamer installe des polices sans sérif pour les mathématiques.

L'option `mathserif` de **beamer** laisse les polices avec sérif en mathématique.

➔ À éviter pour l'harmonie et la lisibilité.

Les environnements `theorem`, `corollary`, `definition`, `definitions`, `fact`, `example`, `examples` sont définis et se présentent comme des blocs.

➔ L'option `notheorems` de **beamer** désactive la définition de ces blocs.

`\newtheorem` et `\theoremstyle` sont disponibles (syntaxe de **amsthm**).

```
\theoremstyle{plain}
\newtheorem{proposition}[theorem]{\translate{Proposition}}
\theoremstyle{example}
\newtheorem{remark}[theorem]{\translate{Remark}}
```

Voir `\translate{-}` plus loin.

Ces environnements acceptent automatiquement un argument `<>` :

```
\begin{definition}<>{-} ... \end{definition}
```

➔ Apparition du bloc en entier.

➔ Insérer des `\uncover<>{-}` à l'intérieur pour affiner les étapes.

Boutons et hyperliens internes

Il est possible et facile de définir des hyperliens internes entre pages.

La syntaxe est celle du *package* **hyperref** :

`\hypertarget<étapes>{label interne}{texte}` définit un but pour un hyperlien interne, de clé interne *label interne*, de but la page PDF résultant de *<étapes>*.

En dehors des spécifications de *<étapes>*, *texte* n'est pas affiché.

`\hyperlink<étapes>{label interne}{texte}` rend *texte* cliquable dans les spécifications *<étapes>* et renvoie vers le but défini par *label interne*.

Pour rendre les choses plus jolies, on peut placer des boutons dans `\hyperlink` :

`\beamerbutton{texte du bouton}` dessine un bouton.

`\beamergetobutton{texte du bouton}` bouton avec un flèche vers la droite.

`\beamerskipbutton{texte du bouton}` bouton avec une double flèche vers la droite.

`\beamerreturnbutton{texte du bouton}` bouton avec une flèche vers la gauche.

```
\hyperlink<3->{labelA}{\beamergetobutton{C'est par là !}}
```

De nombreux buts sont déjà définis par **beamer** auxquels on peut accéder avec les commandes :

`\hyperlinkslidprev{texte}`, `\hyperlinkslidenext{texte}`,

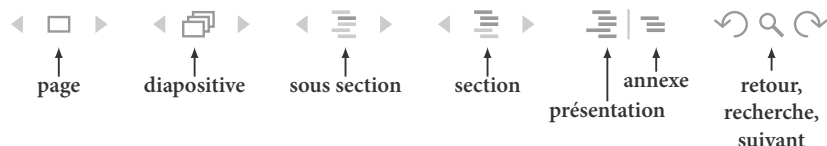
`\hyperlinkframestart{texte}`, `\hyperlinkframeend{texte}`,

`\hyperlinkframestartnext{texte}`, `\hyperlinkframeendprev{texte}`...

La barre de navigation

Par défaut, une barre de navigation est insérée en bas à droite des diapositives.

Elle permet de naviguer dans le document grâce à des hyperliens internes.



page est une page PDF,

diapositive symbolise l'ensemble des pages qui composent une unique diapositive,

sous section est un début de sous section,

section est un début de section,

présentation désigne la fin de la présentation (avant l'annexe),

annexe est le début de l'annexe.

retour, recherche et suivant permettent de naviguer directement dans le fichier PDF.

Les flèches à droite et à gauche des symboles font naviguer d'une entité à une autre.

On peut ôter cette barre en insérant dans le préambule :

```
\setbeamertemplate{navigation symbols}{}%
```

Quelques trucs en vrac

Il est possible d'écraser verticalement une diapositive qui serait trop longue :

<code>\begin{frame}[squeeze]{Titre}</code> Texte un peu trop long <code>\end{frame}</code>	<code>\begin{frame}[shrink=5]{Titre}</code> Texte un peu trop long <code>\end{frame}</code>
--	---

squeeze réduit certains espaces verticaux à zéro.

shrink réduit globalement le contenu de la diapositive du pourcentage donné.

➔ Le contenu de la diapositive est réellement réduit, y compris la taille des caractères !

Il est possible d'interrompre une diapositive, d'en insérer d'autres, puis de la continuer :

```
\begin{frame}<1-5>[label=diapo]{Titre}
  Contenu total
\end{frame}
autres diapositives
\againframe<6->{diapo}
```

La première diapositive contient *tout* le matériel.

D'abord, elle n'est présentée que pour les valeurs `beamerpauses=1` à `5`.

Plus loin, on termine cette diapositive à partir de `beamerpauses=6`.

➔ Possible d'utiliser plusieurs fois `\againframe<>{-}` pour la même diapositive, y compris pour des valeurs de `beamerpauses` se chevauchant.

Quelques trucs en vrac (suite)

Ajouter automatiquement une table des matières à chaque début de section :

```
\AtBeginSection[] {
  \begin{frame}
    \frametitle{\insertsection}
    \tableofcontents[currentsection,hideothersubsections]
  \end{frame}
}
```

`\AtBeginSection` installe du matériel à chaque début de section.

➔ Possible d'ajouter aussi une diapositive avec `\sectionpage`.

Il existe aussi `\AtBeginPart`, `\AtBeginSubsection` et `\AtBeginLecture`

```
\setbeamercovered{transparent}
```

rend les éléments cachés "transparent", donc prévisualisables par le public.

L'inverse est `\setbeamercovered{invisible}` (défaut).

Par défaut, **beamer** prend en compte le texte entre les environnements `frame`.

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

➔ **beamer** ignore le texte hors des `frame` et `\section`...

Quelques trucs en vrac (suite)

Pour gagner de la place verticalement entre du texte et des formules de mathématique :

```
\divide\abovedisplayskip by 2
\divide\belowdisplayskip by 2
```

Utiliser des flèches claires et nettes (\Rightarrow) et non des flèches mathématiques (\Rightarrow) :

```
\usepackage{pifont,relsize}
\newcommand{\flechedroite}{\raisebox{-0.5ex}{%
{\relsize{2}\ding{217}}}\ }
```

→ puiser dans **pifont** d'autres idées...

Ouvrir le document directement en plein écran :

```
\hypersetup{pdfpagemode=FullScreen}
```

Ceci ne fonctionne pas pour tous les visualisateurs PDF.

On ne distribue pas le code source d'une présentation : ne pas hésiter à utiliser des commandes personnelles pour la mise en page, l'apparence, une signalétique...

→ Faire un fichier à part pour les utiliser régulièrement.



Positionner des images dans la diapositive

Il est possible de placer un graphique en bout de ligne sans perturber la mise en page :

```
Du texte
\hfill\raisebox{-3pt}[0pt][0pt]{\makebox[0pt][r]{GRAPHIQUE}}
Suite du texte
```

Le graphique est au final une boîte de taille nulle collée à droite de la diapositive.

Le premier argument de `\raisebox` joue sur sa position verticale.

⚠ Prévoir de la place sur la partie droite du texte. 🗉

Utile aussi pour couvrir momentanément du texte avec un graphique grâce à `\only`.

Pour des mises en page plus complexes, utiliser un graphique **TikZ**.

→ Possibilité de positionner dans un environnement `tikzpicture` du texte, des formules de mathématique, des graphiques et des liens logiques entre tous ces objets.

→ Possibilité de dynamiser le graphique avec des pauses.

Utiliser les colonnes de **beamer** pour placer du texte et des graphiques en vis à vis.



La traduction des mots clés

beamer est accompagné d'un *package* de traduction nommé **translator** (chargé).

Principe : des dictionnaires thématiques sont proposés dans différentes langues, "English" (défaut), "French", "German"...

Ces dictionnaires proposent des traductions pour des mots clés usuels : mois, chiffres, mathématique (théorème, définition...).

La commande `\translate{-}` s'occupe de traduire des mots à l'aide de ces dictionnaires.

`\translate{Theorem}` → Théorème (en français).

`\translate{July}` → Juillet (en français).

Dans le préambule, on charge les dictionnaires de langue française par

```
\uselanguage{French}
```

Ceci équivaut à

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

On active la langue choisie, dans le préambule ou localement dans le texte (basculement) :

```
\languagepath{French}
```

Des vidéos avec beamer

Le *package* **multimedia**, fourni avec **beamer**, permet d'insérer des vidéos et du son dans des documents PDF.

`\movie[-]{poster}{fichier vidéo}` :

- Le code *poster* représente du texte ou une image (`\includegraphics`). Il symbolise l'élément multimédia sur la page.
- En cliquant sur *poster*, le fichier multimédia *fichier vidéo* se lance.
- Les options gèrent le comportement de la vidéo et l'aspect de *poster*.

⚠ Certains visualisateurs PDF ne sont pas capables de gérer des fichiers multimédia.

```
\usepackage{multimedia}
...
\movie[externalviewer,width=120pt]{%
{\includegraphics[width=120pt]{image.jpg}}%
{video.mp4}}
```

Avec l'option `externalviewer`, le visualisateur PDF délègue au système le soin de trouver le lecteur multimédia adéquat. → Il faut régler ce problème au niveau du système.

La documentation de **multimedia** est insérée dans celle de **beamer**.

Autres fonctionnalités à explorer

TikZ est compatible avec **beamer** (même créateur : Till Tantau).

Il fonctionne avec la commande `\uncover<>{-}`. ➔ Graphiques dynamiques.

Position absolue : avec le *package* **textpos**, il est possible de placer des objets de façon absolue dans une diapositive. ➔ Voir la documentation de ce *package*.

Bibliographie : les commandes `\cite` et l'environnement `thebibliography` sont définis mais **beamer** n'est pas compatible avec `bibtex`.

On peut modifier l'aspect de la bibliographie.

Définitions de commandes : il est possible de (re)définir des commandes et des environnements acceptant l'argument `<étapes>` : `\newcommand<>`, `\renewcommand<>`, `\newenvironment<>`, `\renewenvironment<>`.

⚠ Seul **Adobe Reader** ¹⁵⁹ est 100% compatible avec les fonctionnalités ci-dessous :

Transitions : **beamer** fournit des commandes pour installer des transitions :

`\transblindshorizontal<>[-]`, `\transdissolve<>[-]`,
`\transsplithorizontalin<>[-]`...

Animations : `\animate<>` et `\animatevalue<>{-}{-}{-}` permettent des effets d'animations en "jouant automatiquement" une série d'étapes (pages PDF).

Lire la documentation de **beamer** pour plus de renseignements.

Conseils divers

- La mise en page sous forme de listes est très efficace pour segmenter les idées.
- Le découpage d'une diapositive par étapes est souhaitable pour synchroniser la parole avec le visuel.
➔ Mais éviter les éléments cachés "transparents"...
- Les couleurs :
 - Ne surtout pas abuser de couleurs diverses et variées.
➔ Codifier une fois pour toute le choix des couleurs.
 - Ne pas jouer trop finement avec les effets de couleurs : les vidéoprojecteurs ne sont pas fidèles en matière de rendu des couleurs et des contrastes.
➔ Tester certaines couleurs avant de les adopter (ou de les rejeter).
 - Penser aux daltoniens...
- Si l'arrière plan des diapositives est coloré, prévoir des figures avec fond transparent.
➔ Sinon encadrer l'image pour marquer ses bords.
- Ne pas surcharger une diapositive :
 - dans son contenu ;
 - dans les éléments structurants, informatifs, graphiques et visuels qui la composent.
- Faire la chasse aux polices à empattement, surtout dans le mode mathématique.
➔ Éradiquer les `\mathrm{-}` de toutes les commandes personnelles !
- Souvent inutile de numéroter les équations...

9.3 Modifier l'aspect de sa présentation

Modifier des éléments d'une présentation

Règle : presque tous les éléments d'une présentation **beamer** peuvent être modifiés.

Chaque élément requiert trois types de définition pour être rendu :

- un modèle (*template*), c'est à dire du code pour construire l'élément ;
- une couleur (*color*) ;
- une typographie (*font*).

Certains éléments n'ont pas besoin de ces trois types.

Une notion d'héritage est possible entre éléments (*parent*).

À chaque élément est associé un nom assez explicite :

Éléments globaux : `background`, `title page`, `section page`, `headline`, `footline`,
`navigation symbols`, `section in head/foot`, `page number in head/foot`,
`frametitle`, `framesubtitle`...

Éléments locaux : `itemize item`, `itemize subitem`, `enumerate item`, `enumerate subitem`,
`description item`, `bibliography item`, `button`, `section in toc`, `block begin`,
`block end`, `qed symbol`...

Textes : `normal text`, `structure`, `alerted text`, `example text`, `title`, `subtitle`, `author`,
`institute`, `date`...

On modifie l'aspect d'un élément à l'aide du nom associé et de commandes spécifiques.

Les commandes pour modifier les éléments

`\setbeamertemplate{nom de l'élément}{code}` définit la *template* de *nom de l'élément* par le code `LaTeX code`.

Selon le contexte, des commandes sont fournies pour remplir les *templates* : `\inserttitle`,
`\insertpagenumber`, `\insertsection`, `\insertshortauthor`, `\insertframenummer`,
`\insertframetitle`...

`\setbeamertemplate{nom de l'élément}[nom du modèle]` active une *template* avec un nom prédéfini.

`\setbeamercolor{nom de l'élément}{code}` définit la couleur *nom de l'élément* par une liste `fg=`, `bg=`, `parent=` et/ou `use=` séparés par des virgules.

`\setbeamerfont{nom de l'élément}{code}` définit la couleur *nom de l'élément* par une liste `family=`, `series=`, `shape=`, `size=` et/ou `parent=` séparés par des virgules.

```
\setbeamertemplate{navigation symbols}[only frame symbol]
\setbeamertemplate{itemize items}[square]
\setbeamerfont{title}{size=\LARGE,series=\bfseries}
\setbeamerfont{author}{size=\large,shape=\slshape}
\setbeamerfont{alerted text}{series=\bfseries,shape=\slshape}
```

Les couleurs dans beamer : principes

Une couleur au sens de **beamer** est en réalité un couple de couleurs :

Une couleur d'avant plan qui correspond à la couleur de l'élément lui-même, par exemple à la couleur des lettres d'un texte.

Elle est désignée par `fg` (*foreground*).

Une couleur de fond qui correspond à la couleur d'arrière plan.

Elle est désignée par `bg` (*background*).

Quelques couleurs sont définies par défaut :

Couleur du texte : associée au nom `normal text`.

➔ le texte et le fond de la diapositive.

Couleur de structure : associée au nom `structure`.

➔ commande `\structure`, utilisée et déclinée dans de nombreux éléments.

Couleur d'alerte : associée au nom `alerted text`

➔ commande `\alert` et blocs d'alerte.

Couleur d'exemple : associée au nom `example text`

➔ blocs d'exemples.

Il est possible de définir des couleurs pour les mathématiques différentes de celles du texte !

Utiliser les noms (explicites) `math text`, `math text inlined`, `math text displayed` et `normal text in math text...`

Les couleurs dans beamer : définitions et utilisations

Règle 1 : rien n'empêche d'utiliser les commandes usuelles pour les couleurs.

Règle 2 : les couleurs peuvent être définies avec la syntaxe du *package xcolor* (chargé).

```
\setbeamercolor{normal text}{bg=red!20,fg=black!80}
\setbeamercolor{alerted text}{fg=magenta}
```

La syntaxe `bg=` (sans rien après le signe `=`) désigne une couleur "transparente".

`parent=` permet de faire un héritage de couleurs :

```
\setbeamercolor{maman}{bg=red,fg=black}
\setbeamercolor{fiston}{parent=maman,fg=purple}
```

➔ la couleur de fond est héritée, la couleur d'avant plan est redéfinie.

`use=` permet d'utiliser d'autres couleurs et de les modifier :

```
\setbeamercolor{papa}{use={structure,normal text},
fg=structure.fg!50,bg=normal text.bg!80!red}
```

`\usebeamercolor[fg,bg]{nom couleur}` installe une couleur et compose le texte avec.

L'option permet de n'installer que la couleur d'avant plan ou de fond.

`\usebeamercolor[fg]{fiston} Fiston` ➔ Fiston

Changer certaines dimensions

beamer délègue à **geometry** (chargé) le soin de fixer la taille des diapositives.

On peut agrandir les diapositives :

```
\geometry{papersize={160mm,120mm}}
```

⚠ Conserver le rapport 4×3, sauf usage particulier.

D'autres longueurs peuvent être modifiées avec `\setbeamersize{-}`.

Certaines valeurs modifiables sont :

`text margin left=<dim>`, `text margin right=<dim>` définit les marges du texte (hors des barres de navigation).

`sidebar width left=<dim>`, `sidebar width right=<dim>` définit la largeur des barres de navigation à gauche et à droite.

`description width=<dim>`, `description width of={texte}` définit la largeur des items de description.

Changer la taille des marges :

```
\setbeamersize{text margin left=0.3cm, text margin right=0.3cm}
```

Les couches d'une diapositive

beamer construit une diapositive en 3 couches :

Le canevas de fond : c'est le canevas sur lequel tout le reste est dessiné.

`background canvas` est un modèle, une couleur et une typographie.

Les modèles prédéfinis sont :

`default` installe un rectangle de couleur `background canvas`.

`vertical shading` installe un dégradé vertical où il est possible de désigner 3 couleurs et de fixer le point milieu du dégradé.

La couleur `background canvas` hérite de la couleur `normal text`

Le fond principal : c'est un "fond" installé au dessus du canevas de fond.

➔ prévu pour du matériel de type grille, image...

Les modèles prédéfinis sont `default` (vide) et `grid` (grille).

Prévoir du matériel \LaTeX de taille `\paperwidth`×`\paperheight`.

```
\setbeamertemplate{background}%
{\includegraphics[width=\paperwidth]{image.jpg}}
\begin{frame}[plain] ... \end{frame}
\setbeamertemplate{background}[default]
```

installe une image de fond pour une seule diapositive.

Les objets qui composent la diapositive : c'est le reste du matériel placé au dessus...

Des boîtes de couleur

De nombreux éléments sont construits à l'aide de boîtes définies par **beamer**.

```
\begin{beamercolorbox}[-]{-} ... \end{beamercolorbox}
```

crée une boîte colorée : l'argument obligatoire est une couleur **beamer**.
 ➔ la boîte colore le fond et le texte avec bg et fg respectivement.

Parmi les options on trouve :

wd=(*dim*), dp=(*dim*), ht=(*dim*) désignent la largeur, la profondeur et la hauteur finales.

left, center, right indique l'alignement du texte à l'intérieur de la boîte.

shadow, rounded installent un effet d'ombre et des bords arrondis.

ignorebg crée une boîte sans fond.

```
\begin{beamerboxesrounded}[-]{-} ... \end{beamerboxesrounded}
```

crée une boîte colorée, arrondie, avec titre (argument obligatoire, peut être vide).

upper=couleur, lower=couleur désignent les couleurs **beamer** du bandeau de titre et du contenu respectivement.

width=(*dim*) indique la largeur du texte.

shadow= installe (*true*) ou non (*false*) une ombre.

Ces commandes sont utilisables partout dans une présentation.

Quelques personnalisations utiles

Placer le numéro de la diapositive dans le pied :

```
\setbeamertemplate{footline}{%
\begin{beamercolorbox}[ht=0ex,dp=0ex,
leftskip=.3cm,rightskip=.3cm plus1fil]%
{section in head/foot}%
\raisebox{1.125ex}%
{\tiny\bfseries\insertframenumber/\inserttotalframenumber}%
\hfill%
\end{beamercolorbox}}
```

➔ Pour un thème sans footline prédéfini, sinon adapter le modèle existant.

Installer des logos sur la page de titre :

```
\titlegraphic{%
\begin{center}
\includegraphics[height=1.2cm]{LogoCNRS}\hspace{8mm}
\includegraphics[height=1.2cm]{LogoCPT}
\end{center}}
```

➔ Inséré dans le *template title* page par `\inserttitlegraphic`.

Généralité sur les thèmes

Il est fastidieux de redéfinir tous les éléments d'une présentation pour la mettre à son goût.

beamer fournit des thèmes à installer pour modifier tout ou partie des éléments.

Par défaut, un thème appelé `default` est installé (pas nécessaire de le charger).

Les thèmes sont classés en 4 catégories selon ce qu'ils modifient :

Thèmes extérieurs modifient l'aspect des éléments extérieurs de la présentation : entête, pied, barres de navigations, symboles de navigation, titre des diapositives...

`\useoutertheme[-]{-}` dans le préambule charge un tel thème.

Thèmes intérieurs modifient l'aspect des éléments intérieurs de la présentation : listes, blocs, environnements de théorèmes, tables des matières...

`\useinnertheme[-]{-}` dans le préambule charge un tel thème.

Thèmes de couleurs modifient les couleurs des éléments.

`\usecolortheme[-]{-}` dans le préambule charge un tel thème.

Thèmes de fontes modifient les fontes utilisées dans les éléments.

`\usefonttheme[-]{-}` dans le préambule charge un tel thème.

La classe **beamer** accepte les tailles de 8pt à 12pt (jusqu'à 20pt avec **extsize**).

On peut installer des *packages* de polices globaux : **lmodern** (recommandé), **helvet**, **libertine**, **arev**...

Les thèmes extérieurs et intérieurs ne touchent pas aux couleurs et aux fontes.

Les thèmes de couleurs

Il y a trois types de thèmes de couleurs :

Thèmes couleurs extérieurs pour les éléments extérieurs (noms : animaux marins...).

Thèmes couleurs intérieurs pour les éléments intérieurs (noms : fleurs...).

Thèmes couleurs complets pour tout à la fois (noms : animaux volants...).

beamer installe deux palettes de couleurs (*i.e.* une couleur de base déclinée).

palette primary,	palette sidebar primary,
palette secondary,	palette sidebar secondary,
palette tertiary,	palette sidebar tertiary,
palette quaternary,	palette sidebar quaternary.

Ces couleurs n'ont pas de couleur de fond par convention.

Les thèmes extérieurs doivent utiliser `palette primary` et ses déclinaisons.

`primary` correspond aux éléments devant le plus attirer l'attention,

`quaternary` aux éléments devant le moins attirer l'attention.

`palette sidebar primary` et ses déclinaisons sont réservées aux barres de navigations sur le côté de la diapositive.

Un thème couleur extérieur installe en général la palette à partir de la couleur `structure`.

Les thèmes couleurs internes utilisent en plus les couleurs `normal text`, `alerted text` et `example text`. ➔ On peut modifier soi-même ces couleurs.

..... Les thèmes clés en main

Des thèmes de présentation sont aussi définis.

Ils regroupent de façon cohérente des ensembles de 4 thèmes ci-dessus.

`\usetheme[-]{-}` dans le préambule charge un tel thème.

Ces thèmes de présentation portent des noms de villes, sauf `default` et `boxes`.
(Marseille est encore disponible...)

`boxes` est conçu pour ajouter des boîtes dans la barre de navigation et le pied.

```
\usetheme[headheight=2.5ex]{boxes}
\addheadbox{section in head/foot}%
    {\footnotesize\quad\insertsection}
\addfootbox{structure}%
    {\tiny\quad \insertshorttitle, \insertshortauthor}
```

On peut surcharger un thème clés en main :

```
\usetheme{Warsaw}
\usecolortheme{crane}
\setbeamerfont{frametitle}{size=\Large,series=\bfseries}
```

La documentation de **beamer** expose en détail les thèmes possibles.

Le dossier `texmf-dist/tex/latex/beamer/themes/` contient tous les thèmes.

➔ Possible de voir comment ils sont construits pour d'éventuelles modifications.

9.4 Au delà de la présentation

..... La notion de mode

beamer définit 5 modes pour la création d'un document :

`beamer` est le mode par défaut, il correspondant à la présentation.

`second` est le mode utilisé pour une diapositive sur un second écran.

`handout` est une version de la présentation où les pauses sont supprimées.

➔ Possibilité de proposer une version papier et un poster avec **pdfpages**...

`trans` est le mode de création de transparents (pauses supprimées).

➔ Impression sur transparents, prévoir thème adapté.

`article` est un mode où la classe du document n'est pas **beamer**.

➔ Version possiblement élargie de la présentation, avec une autre mise en page.

Le mode `all` les représente tous, le mode `presentation` exclut `article`.

`\mode<mode>{texte}` ne compose *texte* que dans le mode sélectionné.

➔ On peut donc ajouter sélectivement du texte à un des modes.

➔ *texte* peut contenir n'importe quel code \LaTeX : commandes, *packages*...

Certaines commandes acceptent une spécification de mode :

`\section<presentation>{Nom de la section pour les présentations}`

`\section<article>{Nom de la section pour l'article}`

La syntaxe `<>` utilisée pour les pauses accepte aussi les modes :

`\uncover<beamer:2-|article|handout>{texte}`

➔ le texte est toujours présent en modes `article` et `handout`.

..... Préparer la version article

On peut préparer une version `article` en même temps que la version `présentation` :

- Séparer le code source en plusieurs morceaux :

- deux fichiers maîtres pour les modes `beamer` et `article`;

- le contenu de la présentation (`\begin{document} ... \end{document}`).

➔ Fichier `contenu.tex`.

- Baliser `contenu.tex` de commandes `\mode<article>{-}` et `\mode<beamer>{-}`.

Si nécessaire, ajouter des modes dans les `\uncover`, `\only`, `\item`...

```
\documentclass{beamer}
...
\input{contenu.tex}
```

Fichier maître `beamer`

```
\documentclass[12pt]{article}
\usepackage{beamerarticle}
...
\input{contenu.tex}
```

Fichier maître `article`

beamerarticle redéfinit une grande partie des commandes utilisées dans **beamer**.

⚠ Certains environnements ne sont pas redéfinis et sont seulement désactivés :
`frame` et les colonnes par exemple.

Le fichier `contenu.tex` peut contenir plus : *packages*, définitions des commandes...

On peut prévoir d'autres versions, `handout` et `trans`, de la même façon.

.....
Transformer une présentation en poster

Choisir un thème avec seulement les titres des diapositives, pas de numéros de page...
Créer une version handout ➡ handout.pdf

```
\documentclass{article}
\usepackage[alpaper,margin=2cm]{geometry}
\usepackage{graphicx,xcolor}
\pagecolor{blue!10} % couleur de fond
\thispagestyle{empty}
% redéfinir la taille normale du texte

\begin{document}
% code pour le titre, le résumé, les logos

\hfill%
\includegraphics[width=0.45\textwidth,page=1]{handout.pdf}%
\hfill%
\includegraphics[width=0.45\textwidth,page=2]{handout.pdf}%
\hfill\hbox{}
...
\end{document}
```

Cours 10 – Synthèse



10.1 Récapitulatif 204
Où l'on propose un aide-mémoire succinct pour faire le point sur les 9 exposés
passés...

10.2 L'avenir 209
Où l'on se prépare dès maintenant à l'avenir...

10.1 Récapitulatif

Les packages essentiels

Ces packages constituent l'équipement de base obligatoire :

- **etex** pour profiter pleinement du moteur pdf_{tex}...
- **inputenc** et **fontenc** pour l'encodage d'entrée et de sortie...
- **geometry** pour fixer la géométrie du document...
- **babel** pour s'adapter aux langues...
- **textcomp** pour des symboles de texte supplémentaires...
- **lmodern** ou tout autre *package* de polices...
- **hyperref** pour activer les hyperliens et les métadonnées...
- **graphicx** et **xcolor** pour l'insertion des graphiques et l'usage des couleurs...
- **amssymb**, **amsmath** et **mathtools** pour des mathématiques plus belles...
- **natbib** et **bibtex** pour une gestion efficace de la bibliographie...

Des packages utiles

La longue liste de *packages* qui suit permet de résoudre certains problèmes pratiques...

- **amsthm** ou **ntheorem** et **thmtools** gèrent les styles des théorèmes.
- **mathcomp** est le compagnon mathématique de **textcomp**.
- **eucal** et **mathrsfs** fournissent d'autres symboles "script" en mathématique.
- **xy**, **extpfeil**, **12many**, **cancel** et **fouridx** ajoutent des fonctionnalités mathématiques supplémentaires.
- **snapshot** et **bundledoc** collectent les dépendances du document.
- **array**, **tabularx**, **colortbl**, **supertabular**, **longtable**, **hhline**, **multirow** et **booktabs** permettent de mieux composer des tableaux.
- **enumitem** et **paralist** permettent de personnaliser les listes.
- **titling** permet de gérer l'aspect de la page de titre.
- **titlesec** et **titletoc** permettent de personnaliser le style du document.
- **fancyhdr** permettent de personnaliser l'entête et le pied du document.
- **lscap** permet de basculer momentanément en mode paysage.
- **soul** permettent de manipuler le texte : souligner, raturer, surligner...
- **setspace** permet d'ajuster l'espace entre les lignes.

Des packages utiles (suite)

- **appendix** permet de personnaliser (un minimum) les annexes.
- **lastpage** permet de récupérer le nombre total de pages du document.
- **numprint** permet de formater les grands nombres selon les conventions de langue.
- **relsize** permet d'augmenter et diminuer la taille du texte de façon relative.
- **showlabels** et **showkeys** affichent les labels attribués aux éléments du texte.
- **listings** permet de présenter du code informatique avec coloration syntaxique.
- **pdfpages** permet d'inclure des pages d'un document PDF externe.
- **asymptote** permet d'insérer du code Asymptote dans le code source.
- **tikz** permet d'enrichir le document d'éléments graphiques époustouffants.
- **empheq** permet d'encadrer des formules de mathématiques.
- **fancybox** définit des boîtes d'encadrement diverses.
- **float**, **subfig**, **subfloat**, **wrapfig** et **caption** permettent de gérer les flottants et leurs légendes.
- **pifont** et **marvosym** apportent des symboles divers et variés.
- **calc** et **ifthen** permettent une programmation aisée et poussée.
- **tocbibind** permet d'insérer la table des matières, la bibliographie et l'index dans la table des matières.

Des packages utiles (suite)

- **draftwatermark** permet d'insérer un texte grisé en fond de page.
- **multicol** définit un environnement riche pour basculer momentanément en plusieurs colonnes.
- **varioref** définit des commandes de type "\ref" qui ajoutent du texte pour préciser la page.
- **cleveref** redéfinit les commandes de type "\ref" qui ajoutent les types des structures référencées : théorèmes, équations, sections, figures...
- **siunitx** fournit un ensemble de commandes pour formater des nombres en notation scientifique et les unités physiques.
- **chemfig**, **mhchem**, **chemexec**, **chemcompounds** et **mychemistry** composent un écosystème très complet pour dessiner des molécules et des réactions chimiques.
- **xspace** permet aux commandes adaptées (qui utilisent \xspace) de ne pas absorber l'espace qui les suit.
- **csvsimple** et **csvtools** permettent de manipuler des fichiers CSV (*Comma Separated Values*) : tableaux, camemberts...
- **multido** définit une commande pour réaliser des boucles sur des nombres (entiers ou réels), des compteurs ou des longueurs.
- **minitoc** crée une table des matières à chaque début de chapitre.

.....
Des packages utiles (suite)

- **endnotes** définit des notes en fin de documents.
- **lettrine** permet d'insérer des lettrines en début de paragraphe.
- **sfmath** installe des fontes sans sérif en mode mathématique.
- **mdframed** définit un environnement d'encadrement qui accepte les sauts de pages.
- **wordlike** simule un document **Word** : polices, mise en page...
- **spreadtab** permet d'utiliser des fonctionnalités de tableurs dans les environnements de tableau...
- **textpos** permet de placer du matériel sur une page de façon absolue.
- **dashrule** définit une commande pour créer des lignes en pointillés.
- **musixtex** est un ensemble de commandes et de polices destinées à composer des partitions de musique avec L^AT_EX.
- **newfile** permet de sauver du texte dans un fichier et de lire des fichiers.
- **microtype** est une interface pour activer des fonctionnalités typographiques avancées de pdftex.
- **fonttable** permet d'afficher tous les caractères d'une police.
- **ESIEEc**v, **curve**, **ecv**, **europ**ecv, **modern**cv, **simple**cv sont des classes pour rédiger des CV.

.....
Un préambule minimal

```
\documentclass[12pt,twoside]{article}% ou autre classe
\usepackage{etex}
\usepackage[utf8]{inputenc}% ou autre encodage...
\usepackage[TS1,T1]{fontenc}
\usepackage[a4paper,hmargin=2cm,vmargin=2cm]{geometry}
\usepackage[english,french]{babel}
\usepackage{textcomp}
\usepackage{lmodern}% ou autre package de police

\begin{document}
...
\end{document}
```

.....
Un préambule pour un article

```
\documentclass[12pt]{article}% ou la classe de la revue
\usepackage{etex}
\usepackage[utf8]{inputenc}% ou autre encodage...
\usepackage[TS1,T1]{fontenc}
\usepackage[a4paper,hmargin=2cm,vmargin=2cm]{geometry}
\usepackage{amssymb,amsmath,mathtools}
%\usepackage{amsthm,thmtools} % si besoin
%\usepackage[all]{xy} % si besoin
%\usepackage{graphicx,xcolor} % si besoin
\usepackage[square,numbers,compress]{natbib}
\usepackage[english]{babel}
\usepackage{textcomp}
\usepackage{lmodern}
\usepackage{hyperref}

\bibliographystyle{plainnat}% ou le style de la revue
%\newtheorem{theorem}{Theorem}[section]% et autres définitions

...
```

.....
Un préambule pour une présentation

```
\documentclass[10pt]{beamer}
\usepackage{etex}
\usepackage[utf8]{inputenc}% ou autre encodage...
\usepackage[TS1,T1]{fontenc}
\usepackage[english,french]{babel}% avec ou sans french
\usepackage{textcomp}
\usepackage{lmodern}% ou autre package de police
%\usepackage{tikz}% si besoin

\usetheme{Rochester}% ou autre thème
\title[Titre de l'exposé]{Titre de l'exposé}
\subtitle[Sous titre de l'exposé]{Sous titre de l'exposé}
\author{Thierry Masson}
\institute[CPT-Luminy]{
  Centre de Physique Théorique\\
  Campus de Luminy, Marseille}
\date{CPT, 6 juin 2011}

...
```

De la documentation à portée de main

Un aide-mémoire pour trouver les aide-mémoire en PDF :

- **L^AT_EX: Structured documents for T_EX** ¹⁶⁰, la référence de toutes les commandes.
- **Comprehensive L^AT_EX symbol list** ³, la référence pour trouver un symbole.
- **The Not So Short Introduction to L^AT_EX** ².
- **L^AT_EX Wikibook** ¹⁶¹.

On utilisera texdoc pour accéder à la documentation des *packages*.

Un aide-mémoire pour appeler la police :

- **The L^AT_EX Font Catalogue** ⁸⁶.
- **A Survey of Free Math Fonts for T_EX and L^AT_EX** ¹⁶².

Un forum pour poser des questions :

- **L^AT_EX Community : le forum** ¹⁶³.

10.2 L'avenir

Le comportement du bon élève (être abstrait)

- Le bon élève abandonne les fichiers (très) anciens qui ont (trop) vieilli.
 - Il repart d'un code source vierge, minimal, moderne, et maîtrisé.
 - Il élimine tout ce qu'il ne connaît pas !
- Le bon élève crée des modèles (*templates*) pour chaque usage.
 - Il prend un modèle minimal comme point de départ.
 - Il préfère des *packages* adaptés plutôt que des bricolages personnels.
 - Il ajoute des *packages* à usage occasionnel en les commentant.
 - ➔ Il se rappelle de leur existence, il ne les active que si nécessaire.
 - Il place les modèles dans le dossier prévu par son logiciel (TeXworks, TeXShop...).
 - ➔ Il les fait évoluer en fonction des besoins et des nouveautés.
- Le bon élève crée des fichiers de commandes.
- Le bon élève préfère `\text{-}` à `\mbox{-}` et surtout à `\mathrm{-}`.
- Le bon élève gère un dossier texmf :
 - il y dépose ses fichiers de commandes ;
 - il y dépose ses bases bibliographiques ;
 - il y dépose les classes et *packages* des revues ;
 - il y dépose des logos, des images et tout fichier à usage récurrent.
- Le bon élève met à jour son installation régulièrement avec `tlmgr`.
- Le bon élève lit, relit et re-relit la documentation des *packages*...

Ce qui nous attend

Que sera L^AT_EX dans quelques années ?

- Installation, maintenance et utilisation :
 - l'installation est de plus en plus aisée ;
 - la gestion des *packages* et leur mise à jour se simplifie ;
 - l'accès à la documentation des *packages* est plus facile ;
 - des logiciels intégrés modernes sont disponibles (TeXworks).
- UNICODE : ce sera l'encodage des documents sources.
- xetex et luatex : ces moteurs commencent à être opérationnels.
 - luatex sera le remplaçant officiel de pdftex dans quelques années.
- bibl_{at}ex et biber : ce seront les outils bibliographiques de demain.
- L^AT_EX3 ? L'Arlésienne finira bien par se montrer.
- Espoirs :
 - meilleure visibilité des *packages* pour trouver celui dont on a besoin ;
 - choix dans une plus grande diversité des polices mathématiques ;
 - fusion de *packages* qui sont complémentaires et très proches ;
 - enrichissement typographiques et de mise en page : ombres, forme des paragraphes...

Rendez-vous dans un an pour faire un bilan !

à suivre...

URLs des liens cités dans le texte

- ¹<http://www.latex-project.org/>
- ²<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
- ³<http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>
- ⁴<http://home.gna.org/latexrefman/>
- ⁵<http://www.ctan.org/>
- ⁶<http://en.wikibooks.org/wiki/LaTeX>
- ⁷<http://www.tug.org/pracjourn/>
- ⁸<http://latex-community.org/>
- ⁹<http://texblog.net/>
- ¹⁰<http://science.thilucmic.fr/spip.php?article5>
- ¹¹<http://www.tug.org/texlive/>
- ¹²<http://www.tug.org/texworks/>
- ¹³http://www.xmlmath.net/texmaker/index_fr.html
- ¹⁴<http://www.gnu.org/software/emacs/>
- ¹⁵<http://www.gnu.org/software/auctex/>
- ¹⁶<http://jabref.sourceforge.net/>
- ¹⁷<http://aspell.net/>
- ¹⁸<http://www.tug.org/mactex/>
- ¹⁹<http://pages.uoregon.edu/koch/texshop/>
- ²⁰<http://aquamacs.org/>
- ²¹<http://bibdesk.sourceforge.net/>
- ²²<http://code.google.com/p/mactlmgr/>
- ²³<http://cocoaspell.leuski.net/>
- ²⁴<http://www.tug.org/protext/>
- ²⁵<http://miktex.org/>
- ²⁶<http://www.texniccenter.org/>
- ²⁷<http://www.winedt.com/>
- ²⁸http://fr.wikipedia.org/wiki/Donald_Knuth
- ²⁹<http://fr.wikipedia.org/wiki/Typographie>
- ³⁰http://fr.wikipedia.org/wiki/Johannes_Gutenberg
- ³¹[http://fr.wikipedia.org/wiki/Offset_\(imprimerie\)](http://fr.wikipedia.org/wiki/Offset_(imprimerie))
- ³²http://fr.wikipedia.org/wiki/Publication_assistée_par_ordinateur
- ³³<http://fr.wikipedia.org/wiki/Web>
- ³⁴<http://fr.wikipedia.org/wiki/Pdf>
- ³⁵<http://fr.wikipedia.org/wiki/Glyphe>
- ³⁶http://fr.wikipedia.org/wiki/Fonte_de_caracteres
- ³⁷http://fr.wikipedia.org/wiki/Police_d%27écriture
- ³⁸[http://fr.wikipedia.org/wiki/Ligature_\(typographie\)](http://fr.wikipedia.org/wiki/Ligature_(typographie))
- ³⁹<http://fr.wikipedia.org/wiki/Serif>
- ⁴⁰[http://fr.wikipedia.org/wiki/Bloc-notes_\(Windows\)](http://fr.wikipedia.org/wiki/Bloc-notes_(Windows))
- ⁴¹<http://fr.wikipedia.org/wiki/Emacs>
- ⁴²[http://fr.wikipedia.org/wiki/Pico_\(logiciel\)](http://fr.wikipedia.org/wiki/Pico_(logiciel))
- ⁴³<http://fr.wikipedia.org/wiki/Textedit>
- ⁴⁴[http://fr.wikipedia.org/wiki/Fraise_\(éditeur_de_texte\)](http://fr.wikipedia.org/wiki/Fraise_(éditeur_de_texte))
- ⁴⁵<http://fr.wikipedia.org/wiki/Word>
- ⁴⁶<http://fr.wikipedia.org/wiki/Openoffice>
- ⁴⁷<http://fr.wikipedia.org/wiki/Indesign>
- ⁴⁸<http://fr.wikipedia.org/wiki/QuarkXPress>
- ⁴⁹<http://fr.wikipedia.org/wiki/Scribus>
- ⁵⁰<http://fr.wikipedia.org/wiki/Html>
- ⁵¹<http://fr.wikipedia.org/wiki/TeX>
- ⁵²[http://fr.wikipedia.org/wiki/DVI_\(TeX\)](http://fr.wikipedia.org/wiki/DVI_(TeX))
- ⁵³<http://fr.wikipedia.org/wiki/PostScript>
- ⁵⁴<http://fr.wikipedia.org/wiki/Pi>
- ⁵⁵<http://fr.wikipedia.org/wiki/Unicode>
- ⁵⁶<http://www.luaotex.org/>
- ⁵⁷<http://www.lua.org/>
- ⁵⁸<http://fr.wikipedia.org/wiki/TrueType>
- ⁵⁹<http://fr.wikipedia.org/wiki/OpenType>

⁶⁰<http://www.stixfonts.org/>
⁶¹http://fr.wikipedia.org/wiki/Portable_Network_Graphics
⁶²http://fr.wikipedia.org/wiki/Encapsulated_PostScript
⁶³<http://asymptote.sourceforge.net/>
⁶⁴<http://www.ctan.org/search.html>
⁶⁵<http://ams.org/publications/authors/tex/amslatex>
⁶⁶<https://authors.aps.org/revtex4/>
⁶⁷<http://www.springer.com/authors/>
⁶⁸<http://river-valley.tv/classes-styles-conflicts-the-biological-realm-of-latex/>
⁶⁹[http://fr.wikipedia.org/wiki/Point_\(unité\)](http://fr.wikipedia.org/wiki/Point_(unité))
⁷⁰http://fr.wikipedia.org/wiki/Point_pica
⁷¹http://fr.wikipedia.org/wiki/Point_Didot
⁷²http://fr.wikipedia.org/wiki/Unité_Cicéro
⁷³<http://www.ams.org/msc/>
⁷⁴<http://publish.aps.org/PACS/>
⁷⁵http://fr.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange
⁷⁶<http://fr.wikipedia.org/wiki/Windows-1252>
⁷⁷<http://fr.wikipedia.org/wiki/MacRoman>
⁷⁸http://fr.wikipedia.org/wiki/ISO_8859-1
⁷⁹<http://www.unicode.org/>
⁸⁰<http://fr.wikipedia.org/wiki/Tengwar>
⁸¹http://fr.wikipedia.org/wiki/Le_Seigneur_des_anneaux
⁸²<http://www.unicode.org/fr/charts/>
⁸³<http://www.cairn.info/revue-document-numerique-2002-3-page-51.htm>
⁸⁴<http://www.tug.org/pracjourn/2006-1/robertson/>
⁸⁵<http://www.tug.org/fontname/html/index.html>
⁸⁶<http://www.tug.dk/FontCatalogue/>
⁸⁷http://www.ctan.org/tex-archive/info/Free_Math_Font_Survey/survey.html
⁸⁸http://fr.wikipedia.org/wiki/Le_Prisonnier
⁸⁹<http://www.gust.org.pl/projects/e-foundry/tex-gyre/>
⁹⁰<http://www.arxiv.org/>
⁹¹<http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf>
⁹²<http://www.ctan.org/tex-archive/info/latex-samples/TitlePages/titlepages.pdf>
⁹³[http://www.amazon.fr/Latex-Graphics-Companion-Michel-Goossens/dp/0321508920/ref=sr_1_1?](http://www.amazon.fr/Latex-Graphics-Companion-Michel-Goossens/dp/0321508920/ref=sr_1_1?ie=UTF8&qid=1294213916&sr=8-1)
⁹⁴<http://www.tug.org/pracjourn/2007-1/beccari/beccari.pdf>
⁹⁵<http://asymptote.sourceforge.net/asymptote.pdf>
⁹⁶<http://marris.org/asymptote/>
⁹⁷<http://sourceforge.net/projects/pgf/>
⁹⁸<http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>
⁹⁹<http://www.texample.net/tikz/examples/>
¹⁰⁰<http://www.tug.org/pracjourn/2007-1/mertz/mertz.pdf>
¹⁰¹<http://math.et.info.free.fr/TikZ/>
¹⁰²http://fr.wikipedia.org/wiki/Image_bitmap
¹⁰³http://fr.wikipedia.org/wiki/Image_vectorielle
¹⁰⁴http://fr.wikipedia.org/wiki/Courbe_de_Bézier
¹⁰⁵<http://fr.openoffice.org/>
¹⁰⁶<http://fr.openoffice.org/docs/draw.html>
¹⁰⁷<http://inkscape.org/>
¹⁰⁸http://fr.wikipedia.org/wiki/Adobe_Illustrator
¹⁰⁹<http://www.xfig.org/>
¹¹⁰<http://dktools.sourceforge.net/fig2vect.html>
¹¹¹<http://tams-www.informatik.uni-hamburg.de/applets/jfig/>
¹¹²<http://ipe7.sourceforge.net/>
¹¹³<http://fr.wikipedia.org/wiki/Photoshop>
¹¹⁴<http://fr.wikipedia.org/wiki/CorelDraw>
¹¹⁵<http://skipproject.org/>
¹¹⁶http://en.wikipedia.org/wiki/Comparison_of_vector_graphics_editors
¹¹⁷<http://www.aviary.com/>
¹¹⁸<http://fr.wikipedia.org/wiki/Mathematica>

¹¹⁹<http://fr.wikipedia.org/wiki/Maple>
¹²⁰<http://fr.wikipedia.org/wiki/Matlab>
¹²¹<http://www.scilab.org/>
¹²²<http://maxima.sourceforge.net/>
¹²³<http://www.gnu.org/software/octave/>
¹²⁴<http://magma.maths.usyd.edu.au/magma/>
¹²⁵<http://www.sagemath.org/>
¹²⁶<http://www.gnuplot.info/>
¹²⁷http://fr.wikipedia.org/wiki/Microsoft_Excel
¹²⁸<http://fr.openoffice.org/docs/Calc.html>
¹²⁹<http://www.geogebra.org/cms/>
¹³⁰<http://db-maths.nuxit.net/CaRMetal/>
¹³¹<http://www.tug.org/metapost.html>
¹³²<http://en.wikipedia.org/wiki/Matplotlib>
¹³³<http://www.python.org/>
¹³⁴http://fr.wikipedia.org/wiki/Léonard_de_Vinci
¹³⁵<http://www.ctan.org/tex-archive/info/epslatex/english/epslatex.pdf>
¹³⁶<http://www.bibtex.org/>
¹³⁷http://www.ctan.org/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf
¹³⁸<http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html>
¹³⁹<http://www.sciencedirect.com/>
¹⁴⁰<http://www.springerlink.com/>
¹⁴¹<http://www.ams.org/mathscinet/>
¹⁴²<http://www.slac.stanford.edu/spires/>
¹⁴³[http://en.wikipedia.org/wiki/RIS_\(file_format\)](http://en.wikipedia.org/wiki/RIS_(file_format))
¹⁴⁴<http://www.endnote.com/>
¹⁴⁵<http://www.mendeley.com/>
¹⁴⁶<http://www.citeulike.org/>
¹⁴⁷<http://www.zotero.org/>
¹⁴⁸<http://www.mozilla-europe.org/fr/firefox/>
¹⁴⁹<http://www.tug.org/TUGboat/Articles/tb24-1/patashnik.pdf>
¹⁵⁰<http://www.wired.com/wired/archive/11.09/ppt2.html>
¹⁵¹http://fr.wikipedia.org/wiki/Microsoft_Office_PowerPoint
¹⁵²http://fr.wikipedia.org/wiki/Microsoft_Office
¹⁵³<http://fr.openoffice.org/docs/impress.html>
¹⁵⁴[http://fr.wikipedia.org/wiki/Keynote_\(logiciel\)](http://fr.wikipedia.org/wiki/Keynote_(logiciel))
¹⁵⁵<http://fr.wikipedia.org/wiki/IWork>
¹⁵⁶<http://www.gust.org.pl/projects/e-foundry/latin-modern>
¹⁵⁷<https://bitbucket.org/rivanvx/beamer/wiki/Home>
¹⁵⁸http://en.wikipedia.org/wiki/List_of_PDF_software
¹⁵⁹http://fr.wikipedia.org/wiki/Adobe_Acrobat
¹⁶⁰http://svn.gna.org/viewcvs/*checkout*/latexrefman/trunk/latex2e.pdf
¹⁶¹<http://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf>
¹⁶²http://www.ctan.org/tex-archive/info/Free_Math_Font_Survey/survey.pdf
¹⁶³<http://www.latex-community.org/forum/>