

# La thèse électronique en LaTeX

Jean Hare

Université Pierre et Marie Curie  
ED Physique en Île-de-France  
Laboratoire Kastler Brossel  
[jean.hare@upmc.fr](mailto:jean.hare@upmc.fr)

22 novembre 2017

## Sommaire

- 1 Quelques choix préalables
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 Hyperref
- 6 Deux packages dédiés
- 7 La bibliographie
- 8 La validation

## Sommaire

- 1 Quelques choix préalables
  - Distribution, moteur, dialecte
  - Intérêt de la classe book
  - Choix de l'éditeur
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 Hyperref
- 6 Deux packages dédiés
- 7 La bibliographie
- 8 La validation

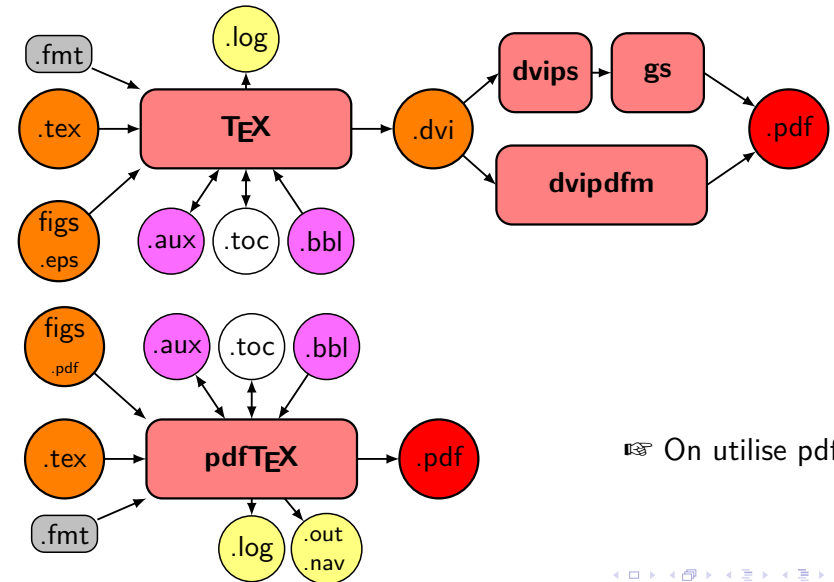
## Distributions TeX

- L'ensemble des programmes et fichiers permettant d'utiliser TeX et ses dérivés sur une système donné sont regroupés en **distributions TeX**, permettant d'avoir une suite logicielle cohérente.
- les distributions TeX ont connu une longue histoire pleine de péripéties ; mais aujourd'hui on a le choix entre MiKTeX (Win\* only), MacTeX (Mac\* only) et TeXLive (toutes plate-formes).
- le choix de TeXLive (inclus dans MacTeX) est tentant mais il a un prix : les fichiers PDF produits sont très souvent invalides ! Sur PC Windows, on privilégiera donc MiKTeX qui n'a pas ce problème.
- Il est important de mettre à jour sa distribution, sous peine de voir surgir des incompatibilités entre packages...  
<http://tex.stackexchange.com/questions/55437/how-do-i-update-my-tex-distribution>

## Moteur T<sub>E</sub>X

- Une distribution T<sub>E</sub>X contient en général une pléthore d'exécutables contenant le mot `tex`, mais la plupart sont des wrappers.
- Les vrais moteurs sont `tex`, `pdftex`, `xetex` et `luatex`.
- Les deux derniers sont des versions modernes présentant deux avantages : prise en charge native de l'unicode et des polices TrueType/OpenType, et production directe de PDF, mais sont significativement plus difficile à prendre en main. De plus LuaTeX est encore en développement...
- Il reste donc à choisir entre les `tex` traditionnel et `pdftex` qui diffèrent essentiellement par la chaîne de compilation et le format des figures.
- Depuis quelques années, même si on fait du TeX traditionnel, le moteur est `pdftex`.

## La chaîne de compilation (simplifiée)



## Choisir un dialecte

- Il n'est évidemment pas question de faire du plain-T<sub>E</sub>X (j'ai fait ça en 1989, je ne m'en souviens que trop bien!).
- Il faut au contraire choisir un système de macros de haut niveau qui soit (1) intelligible et facile à apprendre, (2) extensible, et (3) vous permette de vous concentrer sur le contenu, en laissant l'essentiel du formatage au logiciel.
- Le dialecte L<sup>A</sup>T<sub>E</sub>X, est, à mon avis le seul à répondre de façon immédiate à ces 3 exigences.
- Les puristes vous diront que ConTeXt, développé plus récemment et plus activement, est bien mieux, entre autres parce que beaucoup de fonctions que L<sup>A</sup>T<sub>E</sub>X trouve dans des packages sont intégrées, mais pour moi c'est plutôt un inconvénient. La modularité a bien des avantages...

## Choisir une classe

- On trouve sur CTAN pas moins de 48 packages dont le nom contient le mot `thesis`. Cela tient aux exigences de présentation (layout) de chaque établissement (identité visuelle et choix esthétiques parfois curieux). J'en ai essayé plus d'une dizaine sans jamais être convaincu...
- Il y a aussi des scripts généralistes, comme les classes `memoir` ou `scrbook` de la suite KOMA-Script. Ce sont des sur-ensembles de L<sup>A</sup>T<sub>E</sub>X définissant des centaines de nouvelles commandes et options nécessitant d'apprendre une nouvelle langue. La documentation de KOMA-Script fait 400 pages, celle de `memoir` plus de 600 pages...
- J'ai opté pour utiliser la classe standard `book`, avec les modifications strictement nécessaires pour en améliorer/personnaliser la qualité visuelle, et un minimum de commandes nouvelles. Des packages, qui comportent une documentation courte et ciblée, sont utilisés pour les adaptations les plus délicates.

## Les propriétés spécifiques de la classe book

Par rapport à article ou report cette classe de document implémente un certain nombre de facilités pour un document relativement long :

- option `twoside` par défaut, permettant d'utiliser les commandes ou longueurs `*oddside*` et `*evenside*` ;
- option `\pagestyle{headings}` par défaut, permettant de produire des « running headers » de façon automatique ;
- structuration du corpus avec les commandes `\frontmatter`, `\mainmatter`, `\backmatter`, (numérotation des pages) et aussi `\appendix` (numérotation des chapitres) ;
- définition d'une macro `\part` permettant de scinder le mémoire en parties comprenant plusieurs chapitres.
- jusqu'à 7 niveaux de titres et sous-titre imbriqué dont la numérotation et la présence dans la table des matières sont aisément paramétrables.

## Le layout : pas besoin de de feuilles de style !

L'un des avantages de  $\LaTeX$  réside dans l'aspect professionnel de la typographie, l'homogénéité naturelle des styles si on utilise les commandes de sectionning :

```
1 \chapter[chaptercourt]{chapterlong}
2   \section[sectioncourt]{sectionlong}
3     \subsection{subsection}
```

à la transcription de ce que ferait l'utilisateur ingénue de M $\$$ -Word :

```
1 \leavemode\cleardoublepage\vspace*{3cm}\raggedright
2 {\Large\bfseries\sffamily Chapitre-chapterlong\hfill\thechapter\ }
3 \vspace*{1.5\baselineskip}
4 {\large\bfseries\hspace{1cm}\thesection--sectionlong \ }
5 \vspace*{\baselineskip}\markboth{chaptercourt}{sectioncourt}
```

propice aux erreurs ou aux fantaisies sur l'espacement, les polices, etc. Tant que l'université ne nous impose pas un formatage spécifique, les commandes  $\LaTeX$  suffisent pour assurer l'homogénéité ; nous proposerons néanmoins quelques méthodes de personnalisation, optionnelles.

## Le réglage de l'espacement

- $\LaTeX$  utilise des algorithmes sophistiqués pour le réglage des espacements entre les lettres, mots, lignes paragraphe ou titres, ou pour gérer les veuves et orphelins, notamment grâce à l'utilisation de longueur élastiques, et à l'évaluation mathématique de la beauté d'un paragraphe ou d'une ligne.
- Le résultat n'est pas toujours conforme à ce que vous attendez ou à ce que vous auriez préféré qu'il fasse. Toutefois :

S'abstenir de jouer avec les `\vspace*`, `\newpage`, `\enlargethispage{}` et autres `[!h]` avant l'édition ultime de la thèse, car la moindre modification va casser ces ajustements, résultant en un gâchis considérable de temps.

- Mon conseil : sauf pour les formules mathématiques qu'il faut vérifier au fur et à mesure, ne pas compiler plus d'une fois par heure !

## Le choix de l'éditeur

Le choix de votre éditeur ou d'un IDE  $\TeX$ . est une question de goût et de couleurs. Il y a tout de même des standard de fait :

- Sur Mac\* : **TeXShop** (installe aussi **TeXLive**).
- Sous Unix\* : **Kile**. a longtemps été le standard. **TeXMaker** est sans doute aujourd'hui la meilleure option, mais d'autres préfèrent **emacs** ou **vim** ...

Sous Win\*, plus d'options, car aucun ne surpasse tous les autres :

- **WinEdt** : sans doute la «Rolls Royce» des IDE, mais assez lourd, «trop» personnalisable et a longtemps manqué de la recherche inverse en PDF et du support de l'Unicode.
- **TeXnicscenter**(v.2) et **TeXMaker**, sont multiplateforme et ont un un preview pdf intégré avec recherche directe et inverse.
- **TeXworks** clone multiplateforme de **TeXShop**, en développement actif, très extensible (macros)

Et des outsiders (plus ou moins WYSIWYG) prétendant vous économiser

l'apprentissage de Latex : **BaKoMa**, **Lyx**, **Scientific Word**, etc...

## Le choix de l'éditeur (suite)

En bref, choisir l'éditeur qui vous convient le mieux selon qu'il présente de l'auto-complétion, des palettes de symboles cliquables ou au contraire des raccourcis clavier personnalisables...

Ce dont on a besoin dans tous les cas :

- Le support de l'UTF-8,
- un correcteur orthographique pour l'anglais et le français,
- une aide contextuelle (sinon `texdoc <package>`),
- un bouton et/ou un raccourci clavier pour lancer une compilation,
- la possibilité d'utiliser `Synctex`,
- une analyse de la console de sortie donnant accès direct aux erreurs,
- le moyen d'éditer la ligne de commande : ainsi j'utilise constamment :  
`pdflatex -synctex -shell-escape -interaction=nonstopmode $f` **et**  
`pdflatex -initialize &pdflatex mylatexformat.ltx -jobname="$b" $f`

## Sommaire

- 1 Quelques choix préalables
- 2 **Le préambule**
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 Hyperref
- 6 Deux packages dédiés
- 7 La bibliographie
- 8 La validation

## Préambule minimal : classe, caractères, page

### Préambule minimal

```

1 \documentclass[a4paper,11pt]{book}      caractères moyens
2 \usepackage[utf8]{inputenc}            (cf plus bas) ou |latin9|)
3 \usepackage[english,french]{babel}    francisation des libellés
4 %\usepackage[french,english]{babel} si j'écris en anglais
5 \usepackage[T1]{fontenc}              nécessaire pour césure fr
6 \usepackage{lmodern}                  polices vectorielles "EC"
7 % marges larges : indispensable pour la lisibilité.
8 \usepackage[margin=28mm,bindingsoffset=0mm]{geometry}
9 % Deux alternatives raisonnables pour les caractères
10 %\usepackage[mathptmx]                famille Times texte & math
11 %\usepackage[bitstream-charter]{mathdesign} charter texte & math

```

Les 6 lignes en bleu (ou alternatives) concernant la langue et les caractères sont impératives.

## Commentaires magiques

```

1 %!TeX encoding = UTF-8                Comment. spécial : encodage
2 %!TeX program = pdflatex              moteur
3 %!TeX spellcheck = fr_FR              langue (orthographe)
4 \documentclass[a4paper,11pt]{book}    caractères moyens
5 \usepackage[utf8]{inputenc}          (cf plus bas) ou |latin9|)
6 \usepackage[english,frenchb]{babel}  francisation
7 \usepackage[T1]{fontenc}             nécessaire pour césure fr
8 \usepackage{lmodern}                 polices vectorielles "EC"
9 % marges larges : indispensable pour la lisibilité.
10 \usepackage[margin=28mm,bindingsoffset=10mm]{geometry}

```

### Autres %!

% !TEX root = voir plus bas

% !BIB TS-program =

**Variantes** TeXShop : % !TEX encoding = UTF-8 Unicode

WinEdt : % !Mode: : "TeX:UTF-8" voir tous les modes et submodes

Emacs ou WinEdt : % -\*- coding: UTF-8; -\*-

vim : set fenc=utf-8

## Suite : graphiques et maths

```

11 %% Graphiques :
12 \usepackage{graphicx,color}    inclure graphiques, couleurs
13 \usepackage{svgnames}{xcolor}  noms des couleurs dans SVG
14 \renewcommand{\topfraction}{0.5} autorise 1/2 page de graph. en haut
15 \renewcommand{\bottomfraction}{0.3} et 1/3 page en bas de page
16 \addto\captionsfrench{\def\figurename{{Fig.}}} francisation
17 \addto\captionsfrench{\def\tablename{{Table}}}
18 \addto\captionsfrench{\renewcommand{\CaptionSeparator}{\ \slshape}}
19 %\usepackage{pdfpages}        inclure un PDF (article...)
20 %\usepackage{tikz}usetikzlibrary{...}    dessins intégrés
21 %% Pour toutes sortes de mathématiques :
22 \usepackage{amsmath, mathtools} équations, matrices, etc..
23 \usepackage{amssymb, amsfonts}  tous les symboles math de AMS
24 \usepackage{bm, bbm}            lettres math gras et blackboard
25 \usepackage{upgreek}           grecques pour um et pour beta-decay
26 \usepackage{pifont}           zapfdingbats
27 %\usepackage[overload]{abraces} accolades horizontales
28 %\usepackage[e]{esvect}        flèches plus élégantes

```

## Préambule : utilitaires et personnalisation des entêtes

```

29 %% Utilitaires varies
30 \usepackage{etoolbox}          fonctions avancées pour perso.
31 \usepackage{calc}             calcul infix des longueurs
32 %\usepackage{icomma}          util. virgule comme séparateur décimal
33 \usepackage{versions}         activer ou non certains environnements
34 %% Entetes de page
35 \pagestyle{headings}          chap=>paires section=>impaires
36 %% Personnalisation : remplacement de la capitalstaion au profit de slantsc
37 \usepackage{slantsc}          smallcaps obliques
38 \makeatletter
39 \patchcmd{\chaptermark}{\MakeUppercase}{\scshape\slshape}{}{}
40 \patchcmd{\sectionmark}{\MakeUppercase}{\scshape\slshape}{}{} %
41 \makeatother
42 %% Contrôle plus fin avec les commandes suivantes (déconseillé !) :
43 %\usepackage{fancyhdr}        pour perso manuelle complète
44 %\pagestyle{fancy}           active le style perso. (à définir)
45 %\renewcommand{\headrulewidth}{0pt} ex : suppr ligne sous entête

```

Les personnalisations sont des suggestions...

## Suite : perso. des titres et numéros

```

46 %% Réglages de différens types de numérotations
47 %\setcounter{secnumdepth}{4}    num de chapter àsubsubsection
48 %\renewcommand{\thechapter}{\Roman{chapter}}    ch. Romains pour chapter
49 %\renewcommand{\thesubsubsection}{\alph{chapter}} lettres pour subsub
50 \numberwithin{equation}{section}
51 \numberwithin{figure}{chapter}
52 \numberwithin{table}{chapter}
53 %\mathtoolsset{showonlyrefs} num seult équ. citées (avec \eqref{ })
54 %% Police et retrait des titres de niveau 1, 2, 3
55 %% ce package est incompatible avec minitocs
56 \usepackage{titlesec}          passe les gros titres en sansserif
57 \titleformat{\chapter}[display]{\Huge\sffamily\bfseries}%
58     {\chaptername~\thechapter}{1ex}{}
59 \titleformat{\section}[hang]{\Large\sffamily\bfseries}%
60     {\rlap{\thesection}}{2em}{}
61 \titleformat{\subsection}[hang]{\large\sffamily\bfseries}%
62     {\rlap{\thesubsection}}{3em}{}
63 %% Une personnalisation plus poussée avec tikz : voir template
64 %% Une alternative àtitlesec pour minitocs : voir template

```

## Un autre intérêt de l'encodage utf8

Préférez-vous lire et écrire :

$$i\hbar\frac{\partial}{\partial t}\vert\psi(t)\rangle = \mathsf{H}(t)\vert\psi(t)\rangle$$

ou

$$i\hbar\frac{\partial}{\partial t}\langle\psi(t)\vert = \langle\psi(t)\vert\mathsf{H}(t)$$

et encore :

$$\langle\alpha\vert\alpha(t)\rangle \propto \sum_{n=0}^{\infty}\frac{(\langle\alpha\vert e^{-i\omega t}\alpha\rangle)^n}{n!} e^{-i\omega t}$$

ou

$$\langle\alpha\vert\alpha(t)\rangle \propto \sum_{n=0}^{\infty}\frac{(\langle\alpha\vert e^{-i\omega t}\alpha\rangle)^n}{n!} e^{-i\omega t}$$

Mais avec T1 et lmodern, il vous évite aussi :

$$\tilde{\text{Anonc}}\tilde{\text{A}}\tilde{\text{C}}\tilde{\text{I}}\tilde{\text{A}}\tilde{\text{C}}\tilde{\text{mentaire}}\text{ en m}\tilde{\text{A}}\tilde{\text{C}}\tilde{\text{canique}}\text{ ondulatoire}$$

## Sommaire

- 1 Quelques choix préalables
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 Hyperref
- 6 Deux packages dédiés
- 7 La bibliographie
- 8 La validation

## Organisation du corpus

On peut mettre le préambule dans un fichier `preamble.tex`. Idem pour les données des pages de couverture. La structure du fichier est alors :

```

1 %! special comments
2 \documentclass[a4paper,11pt]{book}
3 \input{preamble.tex}
4 \begin{document}
5 \frontmatter           %numbering \roman
6 \tableofcontent
7 % chapter* de frontmatter centré et dans la toc
8 \chapter*{Introduction}
9 % some intro text
10 \mainmatter           % page numbering \arabic + reset \page counter
11 \chapter[short title]{very long title} % some content
12 \chapter[other short title]{another very long title} % some content
13 \appendix
14 \chapter{title of appendix A}
15 \backmatter
16 % bibliography
17 \end{document}

```

## Accélérer la compilation

On a deux stratégies complémentaires pour accélérer la compilation :

- 1 Fichier scindé en un fichier maître, nommé par exemple `mythesis.tex` et plusieurs fichiers fils, typiquement un par chapitre, importés avec les commandes `\include` et `\includeonly` (et pas `\input`!).
- 2 Création d'un format personnalisé avec le package `mylatexformat.ltx`, qui évite de recompiler tout le préambule avec tous les fichiers inclus (polices et `pgf/tikz` notamment).

Ces deux stratégies feront apparaître de nouveaux commentaires spéciaux :

- 1 Fichiers fils : `%!TeX root = mythesis.tex` (pour l'éditeur)
- 2 Fichier maître : `%&"mythesis"` (pour le compilateur).

## Structure des fichiers maître et fils

### mythesis.tex

```

1 %&"mythesis"
2 %! other special comments
3 \documentclass[...]{book}
4 \input{...}
5 \endofdump
6 \includeonly{Chap2}
7 \begin{document}
8 \frontmatter
9 \tableofcontent
10 \include{Introduction}
11 \mainmatter
12 \include{Chap1}
13 \include{Chap2}
14 \appendix
15 \include{Appendix A}
16 \backmatter
17 % bibliography
18 \end{document}

```

### ChapN.tex

```

1 %!TeX root = mythesis.tex
2 %! other special comments
3 \chapter[short]{long title}
4 % content

```

Création du format personnalisé :

```
1 mylatexformat.ltx -jobname="$b" $f
```

Outil de compilation dans TeXworks :

```
1 pdflatex -initialize &pdflatex
2 mylatexformat.ltx -jobname="$b" $f
```

Modèle dans le dossier `split`.

## Sommaire

- 1 Quelques choix préalables
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 **Les figures**
- 5 Hyperref
- 6 Deux packages dédiés
- 7 La bibliographie
- 8 La validation

## Quel format ?

Les moyens modernes rendent le lecteur très exigeant !

- La méthode LaTeX : `tex`->`dvi`->`pdf` figures compatibles PS, essentiellement eps et quelque bitmap (selon distribution).
- La méthode pdfLaTeX : `tex`->`pdf` requiert des figures PDF, et gère quelques formats bitmap dont notamment png.
- Astuce : `\usepackage{epstopdf}` convertit les .eps en .pdf à la volée (mais config délicate). Sous Windows **EPS2PDF** est plus commode et gère les polices.
- Dans `\usepackage{graphicx}` ne pas préciser le driver, et dans `\includegraphics{}` ne pas préciser l'extension, ainsi le moteur utilisé choira le format le plus approprié.
- Pour les schémas et les courbes, utiliser un format *vectoriel* et pour les images privilégier pdf , ou png à défaut.
- L'importation d'un jpeg dans un Illustrator pour l'exporter après donne généralement une taille 10 fois plus grande. Utiliser à la place Inkscape ou le script **jpeg2ps**, ou mieux **sam2p** qui est plus flexible.

## Quel logiciel ?

Tout dépend bien sûr de vos habitudes.

### Quelques repères :

- Si vous savez utiliser PGF/TikZ ou PGFPlots pour produire des courbes en LaTeX c'est le must, mais sinon il vaut mieux oublier !
- Pour TikZ : **TpX** et **Gnuplot** exportent un code *exploitable*, mais **QTikz**, **TikzEdt** et **TikZiT** sont plus adaptés pour un meilleur code.
- Pour les schémas vectoriels Illustrator ou **Inkscape** sont les références, à moins de se contenter des frustes mais efficaces **Mayura Draw** ou **Xfig**. M\$-Powerpoint n'est pas fait pour cela.
- Pour les donnée expérimentales, ou le résultats de simulations, les meilleurs graphes sont obtenus avec MATLAB ou Python/matplotlib. Même remarque ici pour M\$\_Excel.
- Pour les images on se tourne souvent vers Photoshop, GIMP et ImageMagick, mais ne pas négliger tout ce que vous pouvez faire dans **ImageJ** ou **IGOR Pro** qui sont des logiciels *scientifiques*.

## Les polices

Le principal problème des figures est celui des polices.

### Problèmes principaux :

- Une police manquante ou mal encodée peut rendre le PDF invalide.
- Police indisponible sur le système et/ou l'imprimante de l'utilisateur : à l'affichage ou à l'impression la police fautive sera remplacée par du Courier, taille 12pt, du plus bel effet.
- Notamment les polices standard de Postscript, et leurs clones M\$\_Office ne sont, par défaut, jamais incorporées.
- Vous essayez d'utiliser des polices de LaTeX par souci d'homogénéité, mais elles ne sont plus disponibles à la fin...

Toujours chercher l'option qui permet d'exporter les polices dans la figure, et vérifier dans les propriétés du PDF obtenu que les polices sont incorporées (embedded subset).



## Les polices : des solutions

- On peut tout exporter en bitmap png :- ( ...
- les polices `lmodern` sont fournies à la fois en Type1 et en OpenType, donc utilisables dans tout logiciel, si on les installe au bon endroit.
- MATLAB et Python incorporent des textes LaTeX autonomes.
- Dans Inkscape, l'extension standard `Rendu>>Formule Latex` utilise des outlines. Mieux, l'extension `TeX text` permet de conserver le caractère éditable des formules LaTeX incorporées.
- Pour incorporer les polices *a posteriori* :

```
gs -I "C:\Program Files (x86)\MiKTeX 2.9\fonts\type1" \
-dCompatibilityLevel=1.4 -dPDFSETTINGS=/ebook \
-dCompressFonts=true -dSubsetFonts=true \
-dNOPAUSE -dBATCH -sDEVICE=pdfwrite \
-sOutputFile=output.pdf -f input.pdf \
-c ".setpdfwrite <</NeverEmbed [ ]>> setdistillerparams"
```

Sur la ligne 1, adapter les chemins de Ghostscript et de la distribution TeX.

- Dans le cas désespérés : `ce post` ou Acrobat Pro ...

## Sommaire

- 1 Quelques choix préalables
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 **Hyperref**
- 6 Deux packages dédiés
- 7 La bibliographie
- 8 La validation

## Améliorer les propriétés du PDF

- Ajouter des métadonnées du PDF (inclus dans `thcover`) :

```
1 \title{Dynamique quantique -- Chapitre 1}
2 \author{Jean Hare}
3 \hypersetup{pdftitle={Dynamique quantique}}
4 \hypersetup{pdfauthor={Jean Hare}}
5 \hypersetup{pdfsubject={Chap. 1 : le principe d'évolution}}
6 \hypersetup{pdfkeywords={Schodinger, Heisenberg, Ehrenfest, Feynman}}
```

- Optimiser les bookmarks et la vue :

```
1 \hypersetup{colorlinks,linkcolor=DarkBlue,pdfdisplaydoctitle=true,
2 pdfpagemode=UseOutlines,bookmarksnumbered=true,bookmarksopen=true}
```

- Créer entrée de toc pour les `\chapter*{}` :

```
1 \phantomsection
2 \chapter*{Remerciements}\label{ch:Remerciements}
3 \addcontentsline{toc}{chapter}{Remerciements}
4 \markboth{Remerciements}{Remerciements}
```

## Sommaire

- 1 Quelques choix préalables
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 Hyperref
- 6 **Deux packages dédiés**
- 7 La bibliographie
- 8 La validation



# Coverpages

On fournit les données dans `thcoverdata.tex` et il met tout en page.

# Versionswitch

Version d'archivage

Version de diffusion

```
\documentclass[... ,archiv]{book}
```

```
\documentclass[... ,diffus]{book}
```

FIGURE 1 – caption : Figure from ref. [124]

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nuncummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula fegugit magna. Nunc eleifend consequat lo-rem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

FIGURE 2 – caption

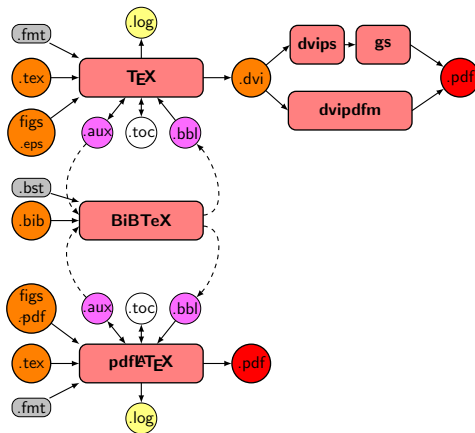
# Sommaire

- 1 Quelques choix préalables
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 Hyperref
- 6 Deux packages dédiés
- 7 **La bibliographie**
- 8 La validation

# La biblio : principe et orientations

- La structure de base est l'environnement : `\begin{thebibliography}{} \bibitem blabla \end{thebibliography}`
- Pour automatiser la production des `\bibitem` on utilise des fichiers `.bib` au format BibTeX (voir [cette documentation générale](#)).
- La génération des `.bib` peut être rendue aisée avec **Zotero** ou une alternative, et leur gestion avec **JabRef**.
- Il y ensuite toute une cascade de choix à faire :
  - Méthode : BibTeX (moteur bibtex) ou BibLaTeX (moteur biber, distribution dépendant). Comme TikZ pour les figures, BibLaTeX est la solution de luxe  $\leftarrow$  BibTeX
  - Styles : la solution de luxe Natbib (qui gère nativement le champ doi et une multitude d'option), ou une multitude d'autre choix plus simples.
- Commandes : `\bibliographystyle{these-fr-href}` % pos. indifférente  
`\bibliography{silica,silicon}` % pos. la biblio

## La biblio : chaîne de compilation



Pour la version finale, couper le cordon entre le latex et les .bib/bst :  
copier le thebibliography du .bbl dans le code latex en remplaçant es  
deux lignes de BibTeX par `\input{mythesis.bbl}`.

## La biblio : Mise en œuvre

- Si vous voulez produire des références conformes à la convention utilisée en physique de façon quasi-universelle soit appel avec [13] et tri dans l'ordre de citation, soit :
  - [13] A. Einstein, "Zur Elektrodynamik bewegter Körper", Ann. Phys. (Berlin) **322**, 891–921 (1905).
 vous pouvez utiliser l'une des 3 versions de `thesesfr**.bst` fournie.
- Si vous avez des URL dans votre fichier BibTeX, les versions avec `-doi` et `-href` vous donneront respectivement :
  - [13] A. Einstein, "Zur Elektrodynamik bewegter Körper", Ann. Phys. (Berlin) **322**, 891–921 (1905). DOI: [10.1002/andp.19053221004](https://doi.org/10.1002/andp.19053221004).
  - [13] A. Einstein, "Zur Elektrodynamik bewegter Körper", [Ann. Phys. \(Berlin\) 322, 891–921 \(1905\)](#).

## Sommaire

- 1 Quelques choix préalables
- 2 Le préambule
- 3 La structure du (ou des) fichier(s)
- 4 Les figures
- 5 Hyperref
- 6 Deux packages dédiés
- 7 La bibliographie
- 8 La validation

## C'est tellement FACILE...

La rédaction terminée, vous pouvez passer à la validation sur FACILE.

**Des problèmes :**

- Avec MiKTeX/pdflatex : 90% des fichiers sont valides ☺
- Avec TeXLive/pdflatex : 90% des fichiers sont invalides ☹

**Des solutions :**

- Toutes plateformes : utiliser `pdftoolkit` en ligne de commande :
 

```
pdftk mythesis_bad.pdf output mythesis_good.pdf
```

 ce qui règle le problème dans la majorité des cas.
- Sous Mac OS X : ouvrir `mythesis_bad.pdf` avec Aperçu PDF et l'enregistrer au format PDF-X.
  - 1 Utiliser l'option `draft` pour savoir si ça vient des figures.
  - 2 Installer `Jhove` pour en savoir davantage.
- Sinon :
  - 3 Utiliser `ghostscript` (commande donnée plus haut ou dans le répertoire script), ou essayer avec Acrobat Pro.
  - 4 Écrire à [jean.hare@upmc.fr](mailto:jean.hare@upmc.fr)