

Document Structure in L^AT_EX

Simon Eveson

Department of Mathematics, University of York

November 2010

Which (free) T_EX distribution?

Windows MikTeX or T_EX Live.

Linux T_EX Live (standard package in Ubuntu, etc.)

Macintosh MacTeX (=T_EX Live + some other bits and pieces)

http:

[//www.tex.ac.uk/cgi-bin/texfaq2html?label=TeXsystems](http://www.tex.ac.uk/cgi-bin/texfaq2html?label=TeXsystems)

http:

[//maths.york.ac.uk/compdoc/LaTeX_on_unsupported_PCs](http://maths.york.ac.uk/compdoc/LaTeX_on_unsupported_PCs)

Which (free) T_EX editor / IDE?

Windows ▶ texworks (basic, bundled with MikTeX)

▶ TeXnicCenter (fancier than texworks)

Linux ▶ kile (standard package in Ubuntu etc.)

Macintosh ▶ TeXShop (bundled with MacTeX)

Anything you like!

Which workflow?

- ▶ For almost all purposes use `pdflatex` which generates PDF directly from \LaTeX source.
- ▶ Use $\text{\LaTeX} \rightarrow .dvi \rightarrow \text{PS} (\rightarrow \text{PDF})$ only in rare circumstances:
 - ▶ You need to include an EPS graphics file.
 - ▶ You're using `\usepackage{pstricks}`.
 - ▶ You need a DVI or PostScript file.
- ▶ If any of the above apply you should preferably:
 - ▶ Convert the EPS file to PDF (e.g. `epstopdf`)
 - ▶ Convert the `pstricks` file to `tikz`.
 - ▶ Ask yourself why!

Source file structure

```
Preamble    \documentclass[class options]{class name}
            \usepackage[package options]{package name} ...
            \newcommand{macro name}{macro definition} ...

            \begin{document}

Topmatter   \title{document title} ... \maketitle

Body        :

            \end{document}
```

The class declaration

```
\documentclass [class options] {class name}
```

- ▶ The class determines the appearance of the document (layout of the title page, size and alignment of section headings, location of page numbers, ...). There are many classes available! Some useful ones:
 - ▶ `article`, `report`, `book`: standard L^AT_EX classes for documents of increasing length.
 - ▶ `amsart`, `amsbook`: originally intended for AMS publications, but generally useful.
- ▶ Class options influence the layout of the text, e.g. size of type, size of paper, single-sided or double-sided, single-column or two-column. Some useful class options:
 - ▶ `10pt`, `11pt`, `12pt` (base font size)
 - ▶ `a4paper`
 - ▶ `oneside`, `twoside`, `onecolumn`, `twocolumn`

With a few exceptions, the document layout can be changed by altering the class and options, with no other editing required.

Package declarations

```
\usepackage [package options] {package name}
```

- ▶ Packages can change document appearance, much like class options. Some useful layout packages:
 - ▶ `mathptmx` (Times/Helvetica/Courier fonts)
 - ▶ `fullpage`, `a4wide` (narrow margins),
 - ▶ `hyperref` (creates internal and external hyperlinks)
- ▶ More commonly, packages add extra features. Some useful packages (others mentioned later):
 - ▶ `tikz` (line diagrams)
 - ▶ `amsmath`, `amsfonts`, `amssymb` (mathematical layout and symbols, automatically loaded by AMS classes)
- ▶ Find many more in the L^AT_EX catalogue: <http://www.tex.ac.uk/tex-archive/help/Catalogue/catalogue.html>
Most of these come as standard in MikTeX, T_EX Live, etc.
- ▶ The distinction between class, class option and package is blurred!

Macro Declarations

`\newcommand{macro name}{macro definition}`
`\DeclareMathOperator{macro name}{operator name}`
(from `amsmath` package, autoloaded by AMS classes)

- ▶ Save typing, improve source readability:

- ▶ `\newcommand{\eps}{\varepsilon}`

- ▶ `\newcommand{\C}{\mathbb{C}}`

- ▶ `\newcommand{\ip}[1]{\left\langle #1 \right\rangle}`

- ▶ `\DeclareMathOperator{\Si}{Si}`

- ▶ `\DeclareMathOperator*{\esup}{ess\,sup}`

$$\begin{array}{c} \varepsilon \\ \mathbb{C} \\ \left\langle \frac{x}{\|x\|}, \frac{y}{\|y\|} \right\rangle \\ \text{Si}(x) \\ \text{ess sup}_{x \in X} f(x) \end{array}$$

- ▶ Many more sophisticated uses.

Personal Macro Files

`\input{filename}`

- ▶ Reads the \LaTeX commands in `filename.tex`
- ▶ Usually looks first the directory / folder containing the master \LaTeX file, then in a local inputs directory — the “local tree” in MiKTeX.
- ▶ Can be used to load a personal collection of macro definitions into a document.
- ▶ Advantages: easy maintenance, instant availability of all personal macros in all documents.
- ▶ Disadvantage: other people (collaborators, journals) can't build your documents without your personal macro file, either as a file or pasted into the document in place of `\input`.
- ▶ Potentially useful package on CTAN <http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=artex>
- ▶ Rough and ready alternative: cut and paste macros from file to file as needed.

Topmatter (I): document information

```
\author{Simon Eveson \and Henning Bostelmann}  
\title{...}  
\date{...}  
:  
\begin{abstract}  
:  
\end{abstract}  
\maketitle
```

- ▶ `\maketitle` goes *after* `\begin{document}` — other macros usually record their arguments for `\maketitle` to typeset later and can go in the preamble or body.
- ▶ Layout depends on document class (e.g. some fields might appear in footnotes, at the end of the document, ...)
- ▶ Fields available, syntax for multiple authors, addresses, email addresses, etc. depend on document class.
- ▶ Read the document class documentation!

Topmatter (II): contents lists

`\tableofcontents`

`\listoffigures`

`\listoftables`

- ▶ Table of contents is built from the sectioning commands `\part`, `\chapter`, `\section`, `\subsection`, etc.
- ▶ List of figures is built from `\begin{figure}... \end{figure}` environments.
- ▶ List of tables is built from `\begin{table}... \end{table}` environments.
- ▶ Numbers and page numbers of sections, figures and tables are shown.
- ▶ \LaTeX sometimes has to be run as many as three times before all page references are correctly resolved. Your IDE will probably take care of this for you.

Master files and `\include`

```
\documentclass{class name}  
\includeonly{file1,file2}  
\begin{document}  
\include{file1}  
\include{file2}  
:  
\include{filen}  
\end{document}
```

- ▶ Break a large document down into several files.
- ▶ The file containing the `\documentclass` declaration and the `\include` and `\includeonly` commands is the *master file*.
- ▶ Use `\includeonly` to process only some of the files.
- ▶ Omit `\includeonly` to process all of the files.
- ▶ Note that each `\include` clears pending floats and starts a new page.

Sectioning (I): section levels

Command	might be typeset as
<code>\chapter{Introduction}</code>	1. Introduction
<code>\section{Definitions}</code>	<i>1.1 Definitions</i>

- ▶ Each documentclass defines a hierarchy of sections.
- ▶ Book-type classes usually have `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` and `\subparagraph`. Article-type classes usually omit `\chapter`.
- ▶ Sections are automatically numbered, as specified by the documentclass: e.g. sections might be numbered 1, 2, 3, ..., subsections 1.1, 1.2, 1.3 ... etc. They can be labelled for cross-referencing.
- ▶ Create unnumbered sections with `\chapter*` etc.
- ▶ Google “latex startsection” if you want to tune the way section headings are displayed (or use a different documentclass).

Sectioning (II): parts and appendices

```
\part{Part Title}
\chapter{Chapter Title}
:
\part{Part Title}
\chapter{Chapter Title}
:
\appendix
\chapter{Appendix Title}
:
```

- ▶ Part numbering is usually independent of chapter numbering: Part 1 might contain Chapters 1–5, Part 2 Chapters 6–10 etc.
- ▶ `\appendix` declares that subsequent sections are appendices. Commonly `\chapter` or `\section` numbering is 1, 2, 3, ... before `\appendix` and A, B, C, ... after `\appendix`

Cross-referencing

```
\label{key}
```

```
⋮
```

```
\ref{key}
```

```
\pageref{key}
```

```
\eqref{key}
```

- ▶ Every automatically generated number (sections, equation numbers, enumerated lists, ...) can be labelled. Put a `\label` command *immediately after* the command generating the number.
- ▶ The label is a private cross-referencing key and is not printed.
- ▶ `\ref` prints the number, `\pageref` the page.
- ▶ `\eqref` is an AMS macro for referencing equation numbers.

Tips for using cross-references

- ▶ Use a tie to stop line breaks: `Section~\ref{key}`.
- ▶ \LaTeX sometimes needs to be run as many as three times before cross-references are correctly resolved. Your IDE will probably take care of this for you.
- ▶ Use descriptive names for labels. A systematic scheme of what the label refers to can be helpful, e.g. `sec:hyperbolic`, `chap:intro`, etc.
- ▶ `\usepackage{showkeys}` prints the keys in the margin of a draft.
- ▶ `\usepackage{smartref}`, `\usepackage{varioref}` and possibly `\usepackage{fancyref}` do lots of clever things between them. Look for documentation on the CTAN:
`www.tex.ac.uk`

HyperTeX

```
\usepackage{hyperref}  
:  
\url{http://www...}
```

- ▶ For use when creating PDF files. Works best with pdf_latex.
- ▶ Table of contents appears as a PDF table of contents.
- ▶ Table of contents entries, cross-references, index entries, etc. become clickable links.
- ▶ References to websites with `\url` become clickable links.
- ▶ For the `\url` macro's formatting features without clickability, `\usepackage{url}`.

Enumeration

```
\usepackage{enumerate}
:
\begin{enumerate}[template]
\item ...
:
\end{enumerate}
```

- ▶ `\item` works with `\label` and `\ref`.
- ▶ `\usepackage{enumerate}` allows you to specify the enumeration labels via the `template` option, e.g.
 - ▶ `[1]` gives 1, 2, 3, ...
 - ▶ `[a]` gives a, b, c, ...
 - ▶ `[(I):]` gives (I):, (II):, (III):, ...
 - ▶ `[Idea 1]` gives Idea 1, IIdeb 2, IIIdec 3, ... so don't get carried away!
- ▶ For finer control, Google "latex enumi"
- ▶ `itemize` is similar but uses bullet points, not numbers.

Floats

```
\begin{float} [options]  
  ⋮  
\caption{Caption Text}  
\label{float label}  
\end{float}
```

- ▶ float can be figure or table
- ▶ \LaTeX moves the float in order to avoid bad page breaks.
- ▶ Make sure the `\label` is *after* the `\caption`. The caption can be at the top or bottom of the float.
- ▶ To stop a float floating, `\usepackage{float}` in the preamble then `\begin{float} [[] H]`. Beware of bad page breaks and floats appearing in a strange order.
- ▶ `\usepackage{caption}` allows you to tune the way the caption is formatted.

Equations

```
\usepackage{amsmath} (not needed in AMS classes)
```

```
⋮
```

```
\begin{equation}
```

```
⋮
```

```
\end{equation}
```

- ▶ Like $$$...$$$ or $\left[...\right]$ but more control over equation numbers (tags)
- ▶ Equations are numbered and work with `\label` and `\ref`.
- ▶ For an unnumbered equation, use `equation*` (or just $$$...$$$ or $\left[...\right]$)
- ▶ To number equations with sections, put something like `\numberwithin{equation}{section}` in the preamble. Equations in section 1 are now numbered 1.2, 1.2, ...
- ▶ To use your own tags, put e.g. `\tag{A}` inside the `equation` environment.

Aligned Equations

`\usepackage{amsmath}` (not needed in AMS classes)

```

\begin{align}
& \vdots \\
x & \&= a + b \\
& \&= c + d \\
& \vdots \\
\end{align}

```

$$\begin{aligned} & \vdots \\ x & = a + b \\ & = c + d \\ & \vdots \end{aligned}$$

- ▶ `&` is the alignment point. `\\` is the end of the line.
- ▶ Each line is numbered using the same counter as equation
- ▶ For an unnumbered alignment use `align*`
- ▶ To suppress numbering on one line, put `\notag` before `\\`
- ▶ To use your own tag, put e.g. `\tag{A}` before `\\`

More on AMS equation layout

- ▶ `amsmath` has many more equation layout features, e.g.
 - ▶ splitting an equation across several lines (`multline`, `split`)
 - ▶ Grouping equations together (`gather`, `gathered`)
 - ▶ Alignments within alignments.

Google “`amsmath` documentation” for the details.

- ▶ AMS classes print tags on the left by default. To print them on the right, use the `reqno` class option.
- ▶ Tags are normally displayed with brackets around them. If you `\ref` an equation number, you just get the number. If you `\eqref` it, you get the brackets as well, in the way the current documentclass prints them.

Theorems (I): declaration and use

```
\usepackage{amsthm} (not needed in AMS classes)
\theoremstyle{style}
\newtheorem{environment name}{theorem name}
:
\begin{environment name} [optional title]
:
\end{environment name}
```

- ▶ Use `\newtheorem` in the preamble, once for each theorem type (lemma, theorem, ...) then `\begin... \end` where required.
- ▶ Styles are `plain`, `example` and `remark`. `plain` style stands out most from the text, `remark` style least.
- ▶ `environment name` is used to refer to the theorem type, `theorem name` is printed; e.g. `\newtheorem{thm}{Theorem}`

Theorems (II): numbering

- ▶ Theorems are numbered and work with `\label` and `\ref`.
- ▶ Several theorem types can share the same numbering sequence, e.g.

```
\newtheorem{thm}{Theorem}
\newtheorem[thm]{lem}{Lemma}
```

Now Theorem 1 will be followed by Lemma 2.

- ▶ Numbering is normally presented as “Theorem 1”. To change this to “1. Theorem.” put `\swapnumbers` in the preamble, *before* the `\newtheorem` declarations.
- ▶ To number theorems within sections, use something like `\newtheorem{prop}{Proposition}[chapter]`
Now, if chapters are numbered I, II, ..., propositions in Chapter I will be numbered I.1, I.2, ...
- ▶ Unnumbered theorems are declared using `\newtheorem*`, e.g.
`\newtheorem*{rem*}{Remark}`
`\begin{rem*} ... \end{rem*}` is then used as usual.

Proofs

```
\usepackage{amsthm} (not needed in AMS classes)
:
\begin{proof}
:
\end{proof}
```

- ▶ Proofs normally begin with the word “Proof.” To change this, use an optional argument, e.g.

```
\begin{proof}[Proof of Cauchy's Theorem]
```

- ▶ End of proof is normally denoted by \square . To change this, redefine `\qedsymbol` in the preamble, e.g.

```
\renewcommand{\qedsymbol}{\blacksquare}
```

finishes proofs with \blacksquare

The bibliography (I): simple use, numeric

```
\cite[extra information]{key1,key2,...}  
:  
\begin{thebibliography}{N}  
\bibitem{key} publication details  
:  
\end{thebibliography}
```

- ▶ `\bibitem` is analogous to `\label`, `\cite` to `\ref`. `key` is a citation key, like a label. It is not printed.
- ▶ `N`: use 9 for 1–9 bibliography entries, 99 for 10–99 etc.
- ▶ The `thebibliography` environment prints a numbered list of its contents, much like `enumerate`
- ▶ `\cite` works in several ways:
 - ▶ `\cite{Eveson2010}` [2]
 - ▶ `\cite{Eveson2009,Eveson2010}` [1,2]
 - ▶ `\cite[Theorem~1]{Eveson2010}` [2, Theorem 1]

The bibliography (II): simple use, author / year

- ▶ For author / year citation, `\usepackage{natbib}` in the preamble. Examples below from the natbib documentation.

- ▶ Bibliography entries now look like

```
\bibitem{Jones et al.(1990)Jones, Baker, and Williams} {jon90} ...
```

- ▶ Jones et al. is the abbreviated citation, 1990 the year and Jones, Baker, and Williams the full citation.
- ▶ Variants of `\cite` work in several ways:

- ▶ `\citet{jon90}` Jones et al. (1990)
- ▶ `\citet[chap. 2]{jon90}` Jones et al. (1990, chap. 2)
- ▶ `\citep{jon90}` (Jones et al., 1990)
- ▶ `\citep[chap. 2]{jon90}` (Jones et al., 1990, chap. 2)
- ▶ `\citep[see] []{jon90}` (see Jones et al., 1990)
- ▶ `\citep[see][chap. 2]{jon90}` (see Jones et al., 1990, chap. 2)
- ▶ `\citet*{jon90}` Jones, Baker, and Williams (1990)
- ▶ `\citep*{jon90}` (Jones, Baker, and Williams, 1990)

BibTeX (I): the basics

```
\bibliographystyle{style name}  
\bibliography{database name}
```

- ▶ BibTeX uses a bibliography database file to create a a bibliography for a particular document, including only the items cited.
- ▶ The bibliography database is a plain text file with extension `.bib` containing a sequence of records.
- ▶ Advantages: ease of maintenance of a single database file, no need to cut and paste or retype citation details into each new document, MathSciNet, SPIRES export in BibTeX format.
- ▶ Possible disadvantage: if you're working on a single large document like a PhD thesis, it might take longer to create the database than to create a single bibliography.
- ▶ Workflow: run \LaTeX , then BibTeX, then \LaTeX once more. Your IDE will probably take care of this for you.

BibTeX (II): a book in the bibliography database

```
@book {MR1084815,  
  AUTHOR = {Horn, Roger A. and Johnson, Charles R.},  
  TITLE = {Matrix analysis},  
  NOTE = {Corrected reprint of the 1985 original},  
  PUBLISHER = {Cambridge University Press},  
  ADDRESS = {Cambridge},  
  YEAR = {1990},  
  PAGES = {xiv+561},  
  ISBN = {0-521-38632-2},  
  MRCLASS = {15-01},  
  MRNUMBER = {1084815 (91i:15001)},  
}
```

BibTeX (III): an article in the bibliography database

```
@article {MR1333035,  
  AUTHOR = {Wiles, Andrew},  
  TITLE = {Modular elliptic curves and {F}ermat's last  
theorem},  
  JOURNAL = {Ann. of Math. (2)},  
  FJOURNAL = {Annals of Mathematics. Second Series},  
  VOLUME = {141},  
  YEAR = {1995},  
  NUMBER = {3},  
  PAGES = {443--551},  
  ISSN = {0003-486X},  
  CODEN = {ANMAAH},  
  MRCLASS = {11G05 (11D41 11F11 11F80 11G18)},  
  MRNUMBER = {1333035 (96d:11071)},  
  MRREVIEWER = {Karl Rubin},  
  DOI = {10.2307/2118559},  
  URL = {http://dx.doi.org/10.2307/2118559},  
}
```

BibTeX (IV): styles

- ▶ `plain`: basic numeric style.
- ▶ `amsplain`: AMS numeric style, like `plain` but with some extra features.
- ▶ `plainnat`: put `\usepackage{natbib}` in the preamble, then use `\cite` for a numeric style and the variant `\cite` commands above for author / year style.
- ▶ Many other styles exist — look on the CTAN.

BibTeX (V): miscellanea

- ▶ There are tools to manage the bibliography database, e.g. JABREF is open-source and multi-platform.
- ▶ BibTeX reformats citation data, often changing case. To protect text, e.g. necessary capital letters, L^AT_EX macros, surround with braces, e.g. {B}anach.
- ▶ BibTeX normally puts only the items you cite into the bibliography. To include an entry without citing it (rarely a good idea!), put `\nocite{key}` before the bibliography. `\nocite{*}` puts the entire database into the bibliography.
- ▶ `\usepackage{citeref}` creates page references from the bibliography to the points of citation.
- ▶ `\usepackage[pagebackref]{hyperref}` does the same thing with hyperlinks.

The index

```
\usepackage{makeidx} (not in AMS classes)
\makeindex
:
\index{topic!subtopic!subsubtopic}
:
\printindex
```

- ▶ `\usepackage` and `\makeindex` go in the preamble; `\index` creates index references from the document body; `\printindex` prints the alphabetically sorted index.
- ▶ The entry will appear in the index under topic, subtopic, subsubtopic. Most commonly only topic is used.
- ▶ To specify a sorting position, use an index entry like `\index{R@ \mathbb{R} }` which puts the symbol \mathbb{R} in the index, sorted as if it were an ordinary R.
- ▶ Workflow: run \LaTeX , then `makeindex`, then \LaTeX . Your IDE will probably take care of this for you.

Websites

- ▶ The CTAN (Comprehensive TeX Archive Network). The principal UK node is `www.tex.ac.uk`.
- ▶ The CTAN home page links to the \LaTeX Catalogue <http://www.tex.ac.uk/tex-archive/help/Catalogue/catalogue.html> — the best place to look for packages.
- ▶ It also links to the UK TeX User's Group (TUG) FAQ <http://www.tex.ac.uk/faq>
- ▶ It also holds the Comprehensive \LaTeX Symbol List <http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>
- ▶ There is a good \LaTeX Wikibook <http://en.wikibooks.org/wiki/LaTeX>
- ▶ The local Mathematics Department page has some useful information <http://maths.york.ac.uk/compdoc/LaTeX>