Modern LaTeX Usage

Thomas Arildsen
tha@es.aau.dk

TPS
Dept. of Electronic Systems
Aalborg University

Licensing and Availability

Source available from GitHub.

PDF slides available from figshare (DOI: 10.6084/m9.figshare.763250).

Appetizer

An attempt to provide some information on best-practice use of LaTeX for typesetting scientific material for reports, articles, presentations, theses etc.
This is not a complete introduction to LaTeX. I assume basic knowledge of using LaTeX in advance. For those interested in an introduction, please see: The Not So Short Introduction to LaTeX.
Some of this advice comes from the document "l2tabu", which I strongly encourage you to read as well.

Agenda

LaTeX Distributions
(La)TeX flavours and workflows
Fonts
Text Encoding
Document Formatting
Mathematics
Numbers and Units
Data
Floating Material
Graphics
Listings
Easy References
Bibliographies
Presentations and Posters
PhD Theses

LaTeX Distributions
TeX Live

TeX Live is a very comprehensive distribution containing more or less everything TeX.

- Cross-platform (Unix/Linux, MacOS, Windows) – comes will all binaries compiled for these platforms.
- Has built-in package manager to update and install additional packages.
- Available here: `http://texlive.org`

LaTeX Distributions
MacTeX

MacTeX is derived from TeX Live but customized for MacOS.

- MacOS only.
- Basically a TeX Live with some extras for MacOS and everything configured for MacOS out-of-the-box.
- Available here: `http://www.tug.org/mactex/`

LaTeX Distributions
MiKTeX

MiKTeX is somewhat like TeX Live a very comprehensive distribution that gives you more or less anything you might need.

- Windows only.
- Has a very clever package manager that will even retrieve missing package on-the-fly while compiling.
- Available here: `http://miktex.org/`

(La)TeX flavours and workflows
TEX

TEX is the original typesetting system developed by Donald Knuth from 1978.

- Perceived by many as being genius.
- Famous for superb typesetting capabilities.
- Currently maintained and latest release from 2008.

(La)TeX flavours and workflows
LaTeX

LaTeX is a macro overlay for TeX that defines a higher-level "language" on top of TeX making it much easier to format documents.

- Developed by Leslie Lamport from the early 1980's.
- The current version is LaTeX 2_ε.
- The newer LaTeX 3 has been under way since the early 1990's.
- Extensive collection of packages for doing more or less anything in LaTeX.

(La)TeX flavours and workflows
pdfTeX/pdfLaTeX

pdfTeX is an extension of TEX with more modern features. pdfLaTeX is its LATEX counterpart.

- Direct compilation to PDF.
- Font handling improvements, e.g. native TrueType and Type 1 font embedding.
- Direct access to PDF features such as hyperlinks, TOC etc.
- *Breaks compatibility with EPS graphics* (no pstricks/psfrag).

(La)TeX flavours and workflows
LuaTeX/LuaLaTeX

LuaTeX is another (newer) extension of TEX.

- Originates from pdfTeX. AFAIK official successor of pdfTex.
- Incorporates scripting in the Lua language.
- Native opentype font support.
- Native unicode.
- Native multi-directional typesetting.
- *Can use system fonts*.

(La)TeX flavours and workflows
XeTeX/XeLaTeX

XeTeX is yet another extension of TEX in a sligthly different direction than LuaTeX.

- Advanced font support with system fonts and special features such as special glyphs, ligatures etc.
- Native unicode.

(La)TeX flavours and workflows

LaTeX → DVI → PostScript → PDF

The classic workflow in LaTeX: LaTeX → DVI → PostScript → PDF.

- Uses the '`latex`' executable to generate a DVI file.
- The DVI file is converted to PostScript using '`dvips`'.
- If you want PDF, you can further convert the PostScript to PDF using, e.g., '`ps2pdf`' or Adobe Distiller.

Originally from the days before anyone thought of PDF.

(La)TeX flavours and workflows
DVI → PDF

More direct path to PDF: LaTeX → DVI → PDF.

- Uses the '`latex`' executable to generate a DVI file.
- The DVI file is converted to PDF using '`dvipdf`'.

Simpler approach if you want PDF but for some reason must go through DVI (e.g., pstricks or psfrag).

(La)TeX flavours and workflows
LaTeX → PDF

The modern approach if you really just want a PDF: LaTeX → PDF.

- Uses the 'pdflatex' executable to generate a PDF file.

It is my impression that many long-time LaTeX users still don't know this option.

Fonts
Choosing different fonts

Handling fonts can be a nightmare in LaTeX. You typically have two options:'

- Stick to LaTeX's default "Computer Modern" font - a Times-like font.
- Become an expert on LaTeX's font handling to customize the use of fonts in your document.

I recommend a third, intermediate solution: use fonts that are available as packages in your system.

Fonts
Font packages

These packages encapsulate all the hard-core mechanics for setting up the font and come with the necessary files (somewhere, don't worry about them).
A good place to start:

- List of fonts with math support and examples shown: http://www.tug.dk/FontCatalogue/mathfonts.html.

Fonts
Font shape

People are often seen using obsolete font commands from earlier versions of LaTeX, such as `{\bf }`, `{\it }` etc. The correct current use of font selection should be:

Shape	Small piece of text	Current environment
Bold	`\textbf{...}`	`\bfseries`
Emphasized	`\emph{...}`	`\em`
Italic	`\textit{...}`	`\itshape`
Medium weight	`\textmd{...}`	`\mdseries`
Roman	`\textrm{...}`	`\rmfamily`
Small caps	`\textsc{...}`	`\scshape`
Sans serif	`\textsf{...}`	`\sffamily`
Slanted	`\textsl{...}`	`\slshape`
Typewriter	`\texttt{...}`	`\ttfamily`
Upright	`\textup{...}`	`\upshape`

Text Encoding
Dealing with “special” characters

Traditionally, in LaTeX you will deal with uncommon characters by using certain commands, for example:

```
\"u \’e \‘e \ae{} \o{} \aa{}
```

Result: ü é è æ ø å
Instead, we can simply choose a text encoding that supports the needed characters, e.g., UTF-8, using the **inputenc** package:

```
\usepackage[utf8]{inputenc}
```

—

```
ü é è æ ø å
```

Result: ü é è æ ø å

Document Formatting
Margins

There is a low-level mechanism to control margins etc. in LaTeX by setting various "lengths".

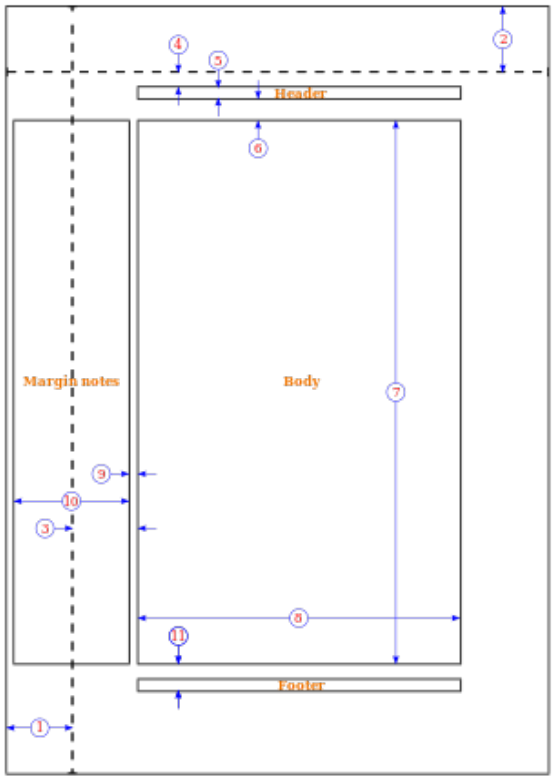

Figure : Figure by Alessio Damato – CC BY-SA 3.0

Document Formatting
Margins

LaTeX page layout can be a real pain to configure.

- Nice solution: the **geometry** package.

```
\usepackage[margin=2cm]{geometry}
```

Just a simple example of 2 cm margin on all sides. Far more details can be specified.

Document Formatting
Line space

LaTeX line spacing can be set at various abstraction levels and can be difficult to set consistently for all elements of the document.

- Again a nice solution: the **setspace** package.

```
\usepackage{setspace}

\singlespacing
\onehalfspacing
\doublespacing
\setstretch{<factor>} % for custom spacing
```

It also provides environments for locally setting the spacing.

Document Formatting I

LaTeX comes with a selection of standard document classes for various purposes, such as: **article, report, book**. These are fine for most purposes.
If you are curious about other document classes, the following are a few examples of high-quality document classes:

KOMA Script Provides the three main classes **scrartcl, scrreprt, scrbook**. Customizable. The page layout of these classes is said to be more "European/A4"-friendly than LaTeX's standard counterparts.

Memoir Also very customizable class. Book-like structure, allowing parts and chapters, but can be used for articles and reports as well. Good for theses.

Document Formatting II

IEEEtran The standard class for most (all?) of IEEE's journals. This is a high-quality article class that you could use for other documents of your own as well as papers submitted to IEEE.

Tufte-latex Provides the classes **tufte-book** and **tufte-handout** by Edward Tufte. A quite different but interesting layout. Check it out if you are a bit adventurous.

Mathematics I

Packages for typesetting mathematics

LaTeX was "born" being excellent for typesetting mathematical symbols, formulae etc. There are however a few useful packages to make your life even easier:

amsmath This package, along with **amsfonts** and **amssymb** provide a host of useful math features. Most notable are its several equation environment building blocks such as **align, aligned, alignat, split, multline, gather, gathered** as well as environments for building matrices and definitions for different cases.

xfrac Provides the command `\sfrac` which prints fractions like this $^1/_2$, which may look better in text than the usual $\frac{1}{2}$.

bm People probably often encounter the problem of making greek-letter variables appear bold, for example to signify matrices or vector. This package provides the solution, the command \bm:

```
$\bm{\epsilon} \bm{\Phi}$
  vs.
$\mathbf{\epsilon} \mathbf{\Phi}$
```

—

$\boldsymbol{\epsilon\Phi}$ vs. $\epsilon\tilde{}$

Mathematics
Other math tips & tricks

- Never use the **eqnarray** environment. Use **amsmath**'s **align** or similar instead.
  For background, see tug.org/pracjourn/2006-4/madsen/madsen.pdf.
- A useful "trick" is to define variable names etc. for your equations as commands, for example:

```
\newcommand{\myVector}{\mathbf x}

$\myVector = \mathbf 0$
```

—

$\mathbf{x} = \mathbf{0}$

This makes it easy to replace the variable name to, say, **y** instead when your supervisor asks you to, all 237 places in the document. Simply edit your command definition!

Numbers and Units

Pretty-printing

The package **siunitx** provides consistent printing of numbers with units.

- As the name suggests, it handles all SI units, but also other units such as bits, bytes etc.

```
\SI{40}{\meter\per\second}
```

—

$40\,\mathrm{m\,s^{-1}}$

- Also handles consistent printing of numbers with customizable precision and many other features.

```
\num[scientific-notation = true,
     round-mode = figures,
     round-precision = 5]{5345.2528592868725}
```

—

5.3453×10^{3}

Data
Displaying data from files

What if you have a comma separated file of data that you want to display? Copy-and-paste it into your source?

- The package **datatool** can actually read data from comma-separated or similar text-based files.
- Lets you build tables from the loaded values.
- Can even plot the data using the auxiliary program **gnuplot**.

Floating Material
Centering floats

It is often seen that people use the **center** environment inside floats (for example **figure, table**) to center the content.

- This may cause unwanted extra vertical space around your figure, table etc.
- Simple fix: use `\centering` instead:

```
\begin{figure}[h]
  \centering
  ...
\end{figure}
```

Floating Material
Sub-figures, -tables etc.

Sometimes you want to collect several figures into one major figure environment ("Fig. 1a, 1b and 1c", for example).

- This can be achieved using the **subfig** package and its `\subfloat` command that wraps the sub-figure, -table etc. inside the containing **figure, table** etc. environment.
- Use **subfig** and not **subfigure**. The latter is outdated.

Floating Material
Wrapping text around a figure

LaTeX places floats so that they occupy the entire horizontal space around them, with no text alongside them. This improves readability and is usually preferred.
If you really want to flow text around a figure, this can be done:

wrapfig — This package lets text wrap around a figure. The figure no longer floats, but text will be placed alongside it.

flowfram — This package lets you do very advanced stuff with text flowing around shapes that can be defined in a drawing program.

Graphics
Displaying from files

LaTeX can display graphics from various file formats. This can be done using the packages **graphics** and **graphicx**.

- Both packages come from the same "family", **graphics** just has a simpler interface than its extended cousin **graphicx**.
- Use these packages instead of **epsf.sty**, **psfig.sty**, **epsfig.sty**; these are outdated.

Graphics
Making LaTeX draw graphics

You can make LaTeX generate graphics according to a script.

- LaTeX has some built-in drawing commands that I will not get into here.
- PSTricks is a package that enables very advanced drawings. Unfortunately, this is based on PostScript and does not work with pdfLaTeX (PDF).
- PGF/TiKZ is a package (or packages: **pgf** and **tikz** – they are two complementary layers of macro languages) that also enables very advanced drawing.
  This works both for DVI → PS and PDF. HIGHLY RECOMMENDED.
- PGF/TiKZ can use a feature to "externalize" graphics, meaning that new drawings are generated on first compilation and subsequently loaded from EPS/PDF files.
  Improves speed and provides stand-alone graphics files.

Graphics
TiKZ, pgfplots and Matlab

PGF/TiKZ have a companion package **pgfplots** that can be used to plot graphs of data very nicely.

- **pgfplots** can be used on its own to set up plots.
- Another option is to use it for plotting figures from Matlab.
  The matlab script "matlab2tikz" can be used to convert Matlab figures to **pgfplots** code that can be rendered in LaTeX using PGF/TiKZ.
  Found here: `http://www.mathworks.com/matlabcentral/fileexchange/22022`
- Huge advantage: all text and numbers in the plots are generated by LaTeX and will automatically match the style of the rest of your document.

Listings I
Algorithms and pseudocode

We often need to list an algorithm or a piece of pseudo-code in a document.

- A useful package for this purpose is **algorithmic**.
- Provides the environment **algorithmic** for typesetting the actual code listing.

```
\begin{algorithmic}
  \FOR{$i=0$ to $10$}
  \STATE carry out some processing
  \ENDFOR
\end{algorithmic}
```

—

for $i = 0$ to 10 **do**
  carry out some processing
end for

Listings II
Algorithms and pseudocode

- The **algorithmic** environment just produces the code text block.
- To create a floating environment (like a table or figure), wrap it in the **algorithm** environment.

```
\begin{algorithm}
  \caption{Some algorithm.}
  \begin{algorithmic}
    \FOR{$i=0$ to $10$}
    \STATE carry out some processing
    \ENDFOR
  \end{algorithmic}
\end{algorithm}
```

—

(Cannot show the **algorithm** float in Beamer. . .)

Listings
Source code

We sometimes need to list actual source code, e.g., Python, C, Matlab etc., as opposed to pseudo-code.

- A useful package for this purpose is **listings**.
- Can display and markup source code from a large selection of languages, including C, Python, Matlab, TeX, LaTeX...
- Can list code pasted in the document or read code from an external file.
- Highly custumizable, can display line numbers, can display excerpts of code.

Listings
Enumerate

This is not really related to source code, but this is nice to know as well.

- You can customize enumerate environments using the **enumerate** package.
- It lets you easily and intuitively change the numbering of items. For example:

```
\begin{enumerate}[2.I]
\item One
\item Two
\item Three
\end{enumerate}
```

—

2.I One
2.II Two
2.III Three

Easy References
Cleveref

Do you ever get tired of keeping track of writing "`Fig.~\ref{...}`", "`Eq.~(\ref{...})`" etc. when cross-referencing in your documents?

- The package **cleveref** can handle this automatically.
- Simply reference using `\cref{...}` in stead of `\ref{...}`.
- Cleveref will automatically figure out what you are referencing and add "Figure", "Table" etc. accordingly.
- Highly customizable in great detail in terms of what to call things; "Figure", "figure", or "Fig." etc.
- References can be converted to plain text if a journal does not support **cleveref**, using the 'poorman' option.

Bibliographies

LaTeX traditionally uses BibTeX to generate bibliographies in documents.

- BibTeX is an old lady and does not dance with unicode text for example.
- The straight-forward fix is to use the "bibtex8" program instead of "bibtex".
  Allows using bibliography files encoded in for example UTF-8 as mentioned earlier so it can handle special characters such as: æ, ø, å, ð etc. typed directly in the bibliography file.

Bibliographies

There is a new bibliography package in town!

- Biblatex is a complete re-design of bibliographies for LaTeX.
- Very customizable.
- Very advanced features (chapter-wise bibliographies, localization. . .)
- Natively handles modern input encodings such as unicode.
- Works together with the backend program “Biber” that sorts the bibliography, instead of “bibtex”.
- Drawback: not supported by IEEE yet.

Presentations and Posters
Beamer

Why use PowerPoint?

- No-one can edit your files—at least not those running Linux ;-)
- You cannot display mathematics properly.

Don't worry. LaTeX is your friend here too.

- Using the package **beamer**, you can easily format slide shows directly in LaTeX.
- Produces PDFs which can be read by everyone, displayed anywhere.
- Beamer has loads of mechanisms to make content change, appear or disappear on slides.

Presentations and Posters
Suggested theme

This presentation was made in Beamer.

- The "theme" used in these slides was created by Jesper Kjær Nielsen from MISP.
- The theme is available here: `http://kom.aau.dk/~jkn/latex/latex.php`.

Presentations and Posters
Poster template

We often need to print posters for presentations at conferences.

- One possibility is the package **baposter**.
- It works by letting you create "boxes" that you fill content into.
- The relative positioning of these boxes can be specified and their sizes are automatically taken care of.
- A couple of the posters in our hallway were made using this package with a theme created by, again, Jesper Kjær Nielsen.
- The theme is available here: `http://kom.aau.dk/~jkn/latex/latex.php`.

PhD Theses
Recommended class

Some of you need to start thinking about writing your thesis soon. . . . ..

- One useful class to do this is **memoir**, mentioned earlier (I used this for my thesis).
- Well-suited for large documents.
- Good-looking layout.
- Easily customizable.
- The previously mentioned KOMA Script bundle should also be very suitable.

PhD Theses
Including papers

Often you will want to do your thesis as a collection of papers. You have all these previously formatted papers that do not fit into your thesis layout. What do you do about them?

- A clever solution is provided by the package **docmute**.
- You include documents using the `\include` command.
- Docmute will automatically remove the preamble of any documents included this way, so they will "obey" the formatting of your master document.

When including bibliographies in your thesis, each included paper will usually have its own bibliography.

- The previously mentioned package **biblatex** can handle this elegantly.
- Using the feature "refsection" or "refsegment", **biblatex** can keep track of what you reference in which section (or segment) and number and list these references for the individual sections.