Preparing Documents with \LaTeX
With a Brief Introduction to Typography

Karl D. Hammond

September 2016
What is \LaTeX?  

• “A Document Preparation System” (macros written in \TeX)  
• \TeX is…  
  • Pronounced as in the Greek “\tau epsilon”, “art,” the root of “technology”  
  • A typesetting program written in 1977–1982 by Donald Knuth (Stanford University Computer Science), author of *The Art of Computer Programming*  
  • Designed to typeset professional-quality documents without a professional typesetter  
  • Designed for typsetting mathematics  
  • Platform-independent (pre-PDF, which was 1993)  
  • Very stable: current version is 3.14159265 → π  

• \LaTeX is…  
  • Written by Leslie Lamport, 1984 (current: \LaTeX 2ε, 1994–2018)  
  • Markup-based (like HTML, say)  
  • Concentrate on content (not presentation)  
  • Logical/structural elements—difficult to write poorly structured documents
**LaTeX vs. Word-Processors**

**Word Processor**
- Designed for **short, one-off documents**
- Format as you type
- What you see is what you get (most of the time)—formatting and content intimately linked
- Line-based
- Easy to get started
- Easy to change local formatting
- Hard to customize/keep things consistent

**LaTeX**
- Designed for **long, multi-section documents**
- Don’t see formatting until the production stage
- Formatting and page layout done separately from content
- Page-based
- Harder to get started
- Hard to predict line breaks, etc.
- Hard to make things inconsistent
\LaTeX vs. Word-Processors

**Word Processor**

- Easy(ish) to customize—but you have to do it repeatedly
- Crashes regularly, particularly with lots of images
- Cross-references are hard
- Citations and bibliographies are hard
- Figures and tables are hard and often mess up page layouts, pagination, and text flow
- Not particularly portable

**\LaTeX**

- Customization is hard, but you only have to do it once
- Very rarely crashes (and even more rarely loses data)
- Cross-references are trivial
- Citations and bibliographies are easy
- Figures and tables are easy and are placed at logical locations by the computer
- As portable as anything gets
Generalized Markup Languages

- Markup languages denote the style or content of text
- Compare these paradigms:
  - “This text is sans serif, size 14, boldface, left-aligned, with 10 pt of space below it, not breaking the page before the following text, with an Arabic numeral 1 in front.”
  - “The enclosed text is a section heading.” (e.g., \section{Introduction})

or perhaps

- “This text should be indented by 0.2 in, justified (flush to the margins), serif, size 10, Roman type, with 0.25 in below and above it but with the text ‘Abstract’ set above it in boldface, centered above the text.”
- “The text between these two delimiters is the abstract.” (e.g.,

\begin{abstract}
Ooh, a lesson in not changing history from Mr.~I’m-My-Own-Grandpa!
\end{abstract}
Getting Started

Input Files

- ASCII text files (*.tex is the traditional file name extension)
- Whitespace is collapsed (e.g., "A    bear" = "A  bear")
- Blank lines are equivalent to \par (paragraph break)
- Comments
- Special characters (# $ % ^ _ & { } ~ \)
- Escape sequences (e.g., \%; exception: \textbackslash)
- Preamble
- Main matter
- Text mode
- Math mode
- Commands (start with a backslash) (e.g., \LaTeX)
- Environments (i.e., \begin{...} and \end{...})
Example 1
A Minimal Document

\documentclass{minimal} \% First line of every file: the document class
\begin{document} \% End of preamble, start of actual text
Some sample text. \% End of document; everything after is ignored
\end{document}

Now make the document into a PDF:

**\LaTeX + DVIPDFm**

$ \texttt{latex example1.tex}$
$ \texttt{dvipdfm example1.dvi}$

**\LaTeX + DVIPostScript**

$ \texttt{latex example1.tex}$
$ \texttt{dvips example1.dvi}$
$ \texttt{ps2pdf example1.ps}$

**\texttt{pdflatex example1.tex}**
Document Classes

Closest MS-Word Analogy: Templates

There are seven “standard” document classes

- **article** For articles in scientific journals, conference proceedings, short reports, etc.; no chapters, just sections
- **book** For actual books
- **letter** For (friendly) letters
- **minimal** Defines as little as possible, just a page size and a `\normalfont` (standard font); used for demos or debugging
- **proc** For proceedings, based on article; intended mostly as a demonstration of how to extend the standard classes
- **report** For longer technical reports containing several chapters; M.S. theses and Ph.D. dissertations fall into this category
- **slides** For slideshows/presentations. Several improvements exist, particularly `beamer.cls` (this presentation)
Document Class Options
Fine-tuning your document

The \texttt{documentclass} command takes several \texttt{optional arguments} that set options without changing the general style. Examples are

- \texttt{10pt,11pt,12pt} Sets default size of main font. Default: 10pt
- \texttt{a4paper,letterpaper,…} Set paper size. Default: letterpaper. Other options: a5paper, b5paper, executivepaper, legalpaper.
- \texttt{fleqn} Equations are left-aligned instead of centered
- \texttt{leqno} Equation numbers are on the left, not the right
- \texttt{titlepage,notitlepage} Specifies whether to create a new page after \texttt{maketitle} (document title) or not. Default: titlepage for report or book and classes derived from them; nomaketitle for article and related documents.
- \texttt{onecolumn,twocolumn} What they sound like
- \texttt{twoside,oneside} Default: twoside for book, but oneside for most other things.
- \texttt{landscape}
- \texttt{openright,openany} Chapters should start either on any page or only on the right-hand (odd-numbered) pages. Irrelevant for article; default is openright in books and openany in report.
Comments

Not just for programmers

Comments are line-based, e.g.,

This is some text. % is it really?
% do we really want this in our company brochure?
Everyone who works for our competitors is an idiot.

This is some text. Everyone who works for our competitors is an idiot.

Comments have the attribute of deleting whitespace at the end and beginning of lines:

This is some% stupid
% better: enlightening
text. My name is Ham% seriously, who
% names himself after salted pork?
mond. % Ha!

This is sometext. My name is Hammond.
Commands

Telling \TeX{} to do a bunch of stuff, possibly behind the scenes

Idea: An interface to tell the software something.

**Syntax**  Start with a backslash, followed by one or more letters, e.g., \texttt{\textbackslash pagestyle}

**Examples**
- \texttt{\textbackslash title} Declare the title of the document
- \texttt{\textbackslash author} Declare the author(s) of the document
- \texttt{\textbackslash maketitle} Make the title page/header

**Arguments** are either the first character after the command or the contents of two braces: \texttt{\textbf{a}} and \texttt{\textbf{\{a\}}} both do the same thing (i.e., \texttt{a}), but \texttt{\textbf{hello}} and \texttt{\textbf{\{hello\}}} are different (\texttt{hello} vs. \texttt{hello})

**Optional Arguments** In brackets, e.g., \texttt{\textbackslash linebreak[3]}
Environments
Telling \TeX{} to treat an entire block of text in a particular way

**Syntax**  Start with `\begin{[name]}` and end with `\end{[name]}`

**Arguments**  Optional arguments can be used: `\begin{figure}[htbp]`

**Use**  Used to surround text and convey document structure

`\begin{abstract}
The real reason people write anything is to prove they are idiots.
\end{abstract}

`\begin{equation}
y = mx + b
\end{equation}`
Line and Page Breaking

- \text{T}_{\text{E}}\text{X} breaks lines automatically and inserts/removes appropriate space\(^1\) to make lines the same length.
- To start a new line \textit{now}, as in an address, use `\textbackslash\textbackslash` or `\texttt{\textbackslash newline}`—this will start a new line without starting a new paragraph.
- To additionally prohibit a \textit{page} break there, use `\textbackslash\textbackslash^*`.
- To start a new page: `\texttt{\textbackslash\textit{newpage}}` or `\texttt{\textbackslash\textit{pagebreak}}`; even better, use `\texttt{\textbackslash\textit{clearpage}}`.
- Page breaks and line breaks have \textit{optional arguments}: `\texttt{\textbackslash linebreak[0]}`, `\texttt{\textbackslash linebreak[1]}`, etc.; `\texttt{\textbackslash linebreak} = \texttt{\textbackslash linebreak[4]}` (pages analogous).
- `\texttt{\textbackslash nolinebreak}` and `\texttt{\textbackslash nopagebreak}` do the opposite\(^2\).
- To make (hideous?) left-aligned text: `\texttt{\textbackslash begin\{flushleft\}}` or `\texttt{\textbackslash raggedright}`.

\(^1\)Typographers call this “kerning”
\(^2\)This only works in paragraph mode (not for headings).
Sectioning Commands

The standard \LaTeX{} classes define the following sectioning commands and associated (default) numbering:

- \texttt{part}: Largest part (think “PART 2” in a long book)
- \texttt{chapter}: (not in articles or proceedings), e.g., 1
- \texttt{section}: typically 1 (articles) or 1.1 (reports/books)
- \texttt{subsection}: typically 1.1 (articles) or 1.1.1 (reports/books)
- \texttt{subsubsection}: typically 1.1.1 (articles) or 1.1.1.1 (reports/books)
- \texttt{paragraph}: unnumbered by default
- \texttt{subparagraph}: unnumbered by default

Each of these has an unnumbered version, e.g., \texttt{\section*{...}}
Sectioning Commands

Change numbering depth with `\setcounter{\secnumdepth}{[value]}`.

Sections (in articles) or chapters (books/reports) can be changed over to alphabetical enumeration (i.e., A, B, C, ...) using the `\appendix` command.

These commands have an optional argument:

```
\section[Short heading]{Long heading that will overflow the margin and annoy someone reading your table of contents because he was just looking for the gist of it}
```
Bulleted and numbered lists in LaTeX are entered similarly:

\begin{itemize}
  \item Hello, world!
  \item Goodbye, cruel world!
\end{itemize}

\begin{enumerate}
  \item Hello, world!
  \item Goodbye, cruel world!
\end{enumerate}

Description lists use the optional argument to \texttt{item}:

\begin{description}
  \item[one] two
  \item[three] four
\end{description}

one two

three four

Changing \texttt{itemize} to \texttt{enumerate} will produce a numbered list.
Typesetting Subtleties

- Line width
Typesetting Subtleties

- Line width
- Non-breaking spaces (~)

...
Typesetting Subtleties

- Line width
- Non-breaking spaces (~)
- The three (some say four) types of dash: hyphens, en dashes, and em dashes (not to be confused with the minus sign)
  - , --, ---, $-1$
  -, −, —, −1
Compare:

10-20 vs. \(10--20\) (ranges);
3-2 vs. \(3-2\) (math mode/minus)

10-20 vs. 10–20 (ranges); 3-2 vs. 3 – 2 (math mode/minus)
Typesetting Subtleties

- Line width
- Non-breaking spaces (~)
- The three (some say four) types of dash: hyphens, en dashes, and em dashes (not to be confused with the minus sign)
  
  - , –, —, $-1$

- , –, —, –1
  Compare:

  10-20 vs. 10--20 (ranges);
  3-2 vs. $3-2$ (math mode/minus)

  10-20 vs. 10–20 (ranges); 3-2 vs. 3 – 2 (math mode/minus)

- Periods at ends of sentences vs. abbreviations vs. initials
Typesetting Subtleties

- Line width
- Non-breaking spaces (~)
- The three (some say four) types of dash: hyphens, en dashes, and em dashes (not to be confused with the minus sign)

$-$, --, ---, $-1$

-, −, —, −1

Compare:

$10-20 \text{ vs. } 10--20$ (ranges);
$3-2 \text{ vs. } 3 -2$ (math mode/minus)

10-20 vs. 10–20 (ranges); 3-2 vs. 3 – 2 (math mode/minus)

- Periods at ends of sentences vs. abbreviations vs. initials
- Quotation marks: ‘‘ vs. ‘’ vs. " becomes “ vs. ” vs. "
Typesetting Subtleties

• Line width
• Non-breaking spaces (~)
• The three (some say four) types of dash: hyphens, en dashes, and em dashes (not to be confused with the minus sign)
  -, --, ---, $-1$
  -, –, —, −1
Compare:

  10–20 vs. \(10--20\) (ranges);
  3–2 vs. \(3-2\) (math mode/minus)

  10-20 vs. 10–20 (ranges); 3-2 vs. 3 – 2 (math mode/minus)
• Periods at ends of sentences vs. abbreviations vs. initials
• Quotation marks: ‘‘ vs. ‘’ vs. " becomes “ vs. ” vs. "
• Ligatures: fill vs. fill or fluffing vs. fluffing (automatic!)
\LaTeX handles accents without having to mess with Character Map, Insert Symbol..., or any other annoyances.

<table>
<thead>
<tr>
<th>è</th>
<th>'e</th>
<th>é</th>
<th>'e</th>
<th>ê</th>
<th>^e</th>
<th>ě</th>
<th>~e</th>
</tr>
</thead>
<tbody>
<tr>
<td>ĕ</td>
<td>=e</td>
<td>ĕ</td>
<td>.e</td>
<td>ě</td>
<td>&quot;e</td>
<td>ç</td>
<td>c c</td>
</tr>
<tr>
<td>ĕ</td>
<td>\u e</td>
<td>ĕ</td>
<td>\v e</td>
<td>ĕ</td>
<td>\H e</td>
<td>ĕ</td>
<td>c e</td>
</tr>
<tr>
<td>Œ</td>
<td>\oe</td>
<td>CE</td>
<td>\OE</td>
<td>æ</td>
<td>\ae</td>
<td>AE</td>
<td>\AE</td>
</tr>
<tr>
<td>å</td>
<td>\aa</td>
<td>Å</td>
<td>\AA</td>
<td>ø</td>
<td>\o</td>
<td>Ø</td>
<td>\O</td>
</tr>
<tr>
<td>l</td>
<td>\l</td>
<td>Ł</td>
<td>\L</td>
<td>l</td>
<td>\i</td>
<td>j</td>
<td>\j</td>
</tr>
<tr>
<td>i</td>
<td>!ˈ</td>
<td>¿</td>
<td>?ˈ</td>
<td>ĩ</td>
<td>\v i</td>
<td>ĭ</td>
<td>\v\i</td>
</tr>
</tbody>
</table>

Se\~nor Martínez está en cárcel.

\textbf{In the T1 encoding (\usepackage[T1]{fontenc})}: \\
č  | \k e
Extensibility
Defining New Commands and Environments

The reason \TeX{} and \LaTeX{} have maintained popularity despite extreme advertising pressure from Micro\$oft and the like is extensibility: a typesetting system designed in 1984 would today have gaps. Because \TeX{} was made to be extensible, those gaps can be closed.

**add-on packages** There are literally dozens of extensions

\begin{verbatim}
\usepackage{txfonts}
\end{verbatim}

**new commands** \begin{verbatim}
\newcommand*{\dosomething}{Do Something}
\end{verbatim}

**re-defined commands** \begin{verbatim}
\renewcommand*{\vec}[1]{\mathbf{#1}}
\end{verbatim}

**new environments** \begin{verbatim}
\newenvironment{itquote}{\begin{quote}\itshape}{\end{quote}}
\end{verbatim}

**new lengths** \begin{verbatim}
\newlength{mylength}, \setlength{mylength}{1.0in}
\addtolength{\floatsep}{0.5\mylength}
\end{verbatim}

This is a large topic—I leave the full exploration of it to you.
\LaTeX \ does \ something \ naturally \ and \ cleanly \ that \ it \ is \ difficult \ to \ get \ MS-Word \ 
to \ do \ \textit{at all}: \ It \ places \ figures, \ tables, \ and \ other \ “floating” \ objects \ in \ a \ logical \ way.

\textbf{Figures}

\begin{figure}
\centering
\includegraphics[width=90mm]{my-graphic}
\caption[This appears in the ToC]{This is the caption, in full-length form. Will appear in the document, formatted accordingly.}
\label{fig:mine} % used to refer to fig. number later.
\end{figure}

Packages related to figures: graphics, graphicx, wrapfig, pdfpages, tikz, pstricks

Packages related to floats in general: float, listings, showlabels
Floating Environments

Figures, tables, and so on

Tables
\begin{table}
\caption{This appears in the ToC}{This is the caption, in full-length form. Will appear in the document, formatted.}
\label{table:mine} % used to refer to table number later.
\centering
\begin{tabular}{l c r p{1in}}
\hline
Left & Centered & Right & Justified \\
\cline{2-3}
blah blah & blah blah & blah blah & blah blah blah \\
\hline
\end{tabular}
\end{table}

Packages related to tables: array, tabularx, dcolumn, longtable, supertabular, booktabs
Mathematics in \LaTeX

This is why you should use \LaTeX! (or \TeX in general)

\TeX has two modes: text mode and math mode.

In \LaTeX, you enter math mode in one of three ways:

\$\ldots\$ or \(\ldots\) Enters in-line math mode, e.g., \((ax^2 + bx + c = 0)\), which produces \(ax^2 + bx + c = 0\).

\[\ldots\] Enters display math mode. Equivalent to the displaymath environment.
Example: \[ ax^2 + bx + c = 0 \]

\(ax^2 + bx + c = 0\)

Environments  Enters display mode, but with numbered equations, e.g.,

\begin{equation}
ax^2 + bx + c = 0
\end{equation}

\(ax^2 + bx + c = 0\)  \(1\)
Math Is a BIG Topic

Some Nuggets

Exponents \((x^2 + y^3)\) is \(x^2 + y^3\)

Subscripts \((x_{ij} \gamma_i P_j)\) is \(x_{ij}\gamma_iP_j\)

Fractions \(\left(\frac{a+1}{b+c}\right)\) is \(\frac{a+1}{b+c}\) (inline mode); \(\left[\frac{a+1}{b+1}\right]\) is \(\frac{a + 1}{b + 1}\)

Parentheses/brackets \(\left[1 + \exp\left(\frac{V}{RT}\right)\right]^2\)

Braces

\[
\delta_{mn} = \left\{ \begin{array}{ll}
1 & m = n \\
0 & m \neq n
\end{array} \right. \]

\[
\begin{array}{l}
1 \ & m = n \\
0 \ & m \neq n
\end{array}
\]
## More Math Nuggets

### Integrals
\[
\int_{0}^{L} \sin\left(\frac{2\pi x}{L}\right) \, dx
\]

### Sums
\[
\sum_{n=0}^{3} x_n = x_0 + x_1 + x_2 + x_3
\]

### Products
\[
\prod_{n=1}^{10} a_n = a_1 a_2 \cdots a_{10}
\]

### Partial Derivatives
\[
dU = \left(\frac{\partial U}{\partial S}\right)_V \, dS + \left(\frac{\partial U}{\partial V}\right)_S \, dV = TdS - PdV
\]

\[
dU = \left(\frac{\partial U}{\partial S}\right)_V \, dS + \left(\frac{\partial U}{\partial V}\right)_S \, dV = TdS - PdV
\]
AMS-LaTeX: The amsmath Package

The amsmath package defines many math-related things that do things like:

- Text-mode containers (\text{...}) for subscripts/superscripts
- Numbering as sub-equations (e.g., Eq. (3a) and (3b))
- Multi-line equations and alignment thereof

\[
x = a + b + c \quad (2)
\]
\[
= a^2 + b^2 \quad (3)
\]
- “Split” equations (one number, multiple lines)

\[
x = a + b + c
\]
\[
= a^2 + b^2 \quad (4)
\]
- Customized brackets/grouping symbols
- Double and triple integrals, line integrals, etc.
- Theorem environments
- … and so on
Tables of Contents

No, wait... this is why you should use \LaTeX.

So you want a table of contents, with all the section numbers automatically updated? You want all the titles to match when you update the section headings? And you want the page numbers to update automatically as you add chapters/sections/etc.? No problem!

Table of Contents \texttt{\textbackslash tableofcontents}

List of Figures \texttt{\textbackslash listoffigures}

List of Tables \texttt{\textbackslash listoftables}

With the \texttt{float} package, any other float, such as a Scheme, can be generated with, for example, \texttt{\textbackslash listof\{schemes\}}

\textit{Updating the Table of Contents requires an extra run of \LaTeX (possibly two).}

Packages for customizing tables of contents without getting “dirty”:\texttt{tocloft, float, titletoc}
Cross-References

Oh, no, I forgot: This is why you should use LaTeX!

Let’s say I have an equation. I define it with the equation environment, and I give it a name by which I can refer to it later:

\begin{equation}
 a \, x^2 + b x + c = 0 \\
\label{eq:quadratic}
\end{equation}

produces

\[ ax^2 + bx + c = 0 \] (5)

Now let’s say I want to refer to it later, as Equation ([number]). I just say Equation~(\ref{eq:quadratic}), and I get “Equation (5).” With the amsmath package loaded, I can even say Equation~\eqref{eq:quadratic}, and I get “Equation (5).”
Citations and Bibliographies

No, seriously, I mean it this time... *this* is why you should use \LaTeX!

Two parts: citations and either bibliographies or reference lists.

Citations  Use the `\cite` command:

This assumes that Hammond and coworkers’ model is correct~`\cite{Hammond 2015a}`.

Bibliographies  Delimited by the `thebibliography` environment.

You have two options:

1. Write `thebibliography` manually
2. Use \BibTeX and the `\bibliographystyle` and `\bibliography` commands to generate your `thebibliography` environment automatically

Bibliography-related packages: natbib, cite, mciteplus, natmove, refcount, biblatex, bibentry, footbib
BibTeX is a language designed to write bibliographies.

- Based on straightforward input files (*.bib)
- Relatively easy to make custom bibliography styles (*.bst) using \texttt{latex makebst}
- Difficult to do something “unusual” (such as boldfacing one’s name in a CV-style bibliography), but it can be done
- Extensible to handle “modern” media (e.g., URLs, DOIs)
- Stack-based (“reverse Polish notation”)
- Authors can match desired bibliography style precisely without knowing what that style is
- Modern replacement: BibLaTeX
BibTEX: The database (*.bib)

Example entry:

```latex
@string{epl = "Europhys.\ Lett."}
string{epl = "Europhysics Letters"}
@article{Hammond2015a,
    author = "Hammond, Karl D. and Hu, Lin and Maroudas, Dimitrios and Wirth, Brian D.",
    title = "Helium Impurity Transport on Grain Boundaries: {E}nhanced or Inhibited?",
    journal = epl,
    volume = 110,
    pages = 52002,
    doi = "10.1209/0295-5075/110/52002",
    year = 2015
}
```

case is ignored for non-string data in BibTEX bibliography databases.
In \texttt{BibTeX}, \textit{comments} are everything \textit{not} between \texttt{@.*{ and the corresponding closing brace.}

Your \texttt{BibTeX} database consists of \textit{entry types}: \texttt{@article, @book, @incollection}, etc.

There are several “standard” \textit{fields} that are defined by bibliography style files, corresponding to different types of \texttt{BibTeX} data, such as \texttt{address}, \texttt{author}, \texttt{booktitle}, \texttt{chapter}, etc.

These fields may be mandatory, optional, or completely ignored for a given entry type in a particular bibliography style.
Example 2: Write an Article in \LaTeX

Technically, we’re going to be taking an already-written paper and using \LaTeX to reformat it.

Things to remember:

• Don’t focus on formatting (yet): just content
• Resist the urge to copy an example from someone else—start from a blank document, and add as you need!
• Don’t add a package to your document unless you know what it does and why you need it!
Example 3: Make Two Articles into Chapters

Oh, sh*t, I’m going to have to say it again: *this* is why you should use \LaTeX!

This is really easy:

1. Strip off the `\documentclass` and `\usepackage` directives
2. Start a “master” document that has chapters in it
3. Replace `\title` with `\chapter` and delete or comment out the `\author`, `\date`, etc. information in the preamble
4. Transfer any relevant `\usepackage` commands to the preamble of the master document
5. Include each chapter with `\include`

---

3 There are packages that allow you to skip even *this* easy step.