An Introduction to LaTeX

Iryna Schlackow
Mathematical Institute

This talk and other useful LaTeX related information is available at
http://www.maths.ox.ac.uk/help/faqs/latex/

November 21, 2011

The name of the game

TEX is a computer program created by Donald E. Knuth. It is aimed at typesetting text and mathematical formulae. Knuth started working on it in 1977, and TEX as we use it today was released in 1982.

TEX is renowned for being extremely stable and virtually bug free. The version number of TEX is converging to π and is now at 3.1415926.

TEX is pronounced "Tech," with a "ch" as in the German "Ach." In an `ASCII` environment, TEX becomes `TeX`.

The name of the game

LaTeX is a TeX macro package. It enables authors to typeset and print their work at the highest typographical quality, using a predefined, professional layout.

LaTeX was originally written by Leslie Lamport in 1980s, and its current version, LaTeX 2_ε, was released in 1994.

LaTeX (`LaTeX` in an ASCII environment) is pronounced "Lay-tech" or "Lah-tech." LaTeX 2_ε (`LaTeX2e`) is pronounced "Lay-tech two e".

Why LaTeX?

- The typesetting of mathematical formulae is supported in a convenient way.
- Complex structures (cross-references, bibliography) can be generated easily.
- Professionally crafted predefined layouts are available, and another document class styles can be easily superimposed.
- Many scientific journals accept manuscripts in LaTeX only.
- The system is free and runs on almost any hardware and software platform available.

Why LaTeX?

LaTeX is built on a programming language and is extensible. There exist many (free!) add-ons:

- customised class styles for scientific journals, theses (`ociamthesis`), presentations (`beamer`) and letters;
- packages for writing CV, typesetting music and linguistic papers and producing coffee stains;
- LaTeX can be integrated with other programs (e.g. *Sweave* combines R and LaTeX);
- ...

Any disadvantages?

- Not (traditionally) a WYSIWYG system.

LaTeX vs WYSIWYG systems

WYSIWYG systems (Word)	LaTeX (traditional approach)
The output is precisely what you type in.	You type in LaTeX "code", which needs to be compiled first.

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	`\left(\begin{array}{cc}`
	`1 & 2 \\`
	`3 & 4`
	`\end{array}\right)`

The document layout is specified by means of *visual design*.	A suitable layout is chosen by LaTeX once the *logical structure* of the document has been specified.
You may spend too much time fiddling with fonts and margins. The document is likely to have little or inconsistent structure.	It is very hard to write unstructured and disorganised documents in LaTeX.

LaTeX vs WYSIWYG systems

So, coming back to our matrix:

$$\left(\begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array}\right)$$

```
$\left(\begin{array}{cc}
1 & 2 \\
3 & 4
\end{array}\right)$
```

This is

- an array with
- 2 columns: all centred;
- & acting as a column separator;
- \\ acting as a row separator;
- surrounded by round brackets (,) of the appropriate size.

LaTeX vs WYSIWYG systems

So, coming back to our matrix:

$$\left[\begin{array}{crl} 11 & 12 & 25 \\ 3 & 4 & 6 \\ 342 & 234 & 232 \end{array}\right]$$

```
$\left[\begin{array}{crl}
11  & 12  & 25 \\
3   & 4   & 6  \\
342 & 234 & 232
\end{array}\right]$
```

This is

- an array with
- 3 columns: centred, right-aligned, left-aligned;
- & acting as a column separator;
- \\ acting as a row separator;
- surrounded by square brackets [,] of the appropriate size.

LaTeX vs WYSIWYG systems

There exist LaTeX-based WYSIWYG and the like programs:

- Scientific Word (for Windows, commercial)
- LyX (Unix, Mac OS X and Windows, free)

Every LaTeX input file possesses a certain structure. You start with specifying what sort of document you intend to write

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

This is followed by the preamble where you can include commands influencing the style of the whole document or load packages adding new features to the LaTeX system

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

When all the setup work is done you start the body of the text

```
\begin{document}
```

Now you enter the text mixed with LaTeX commands. At the end of the document you add

```
\end{document}
```

Anything that follows this command will be ignored by LaTeX.

Once the LaTeX file is compiled, a viewable output file is produced. It can be a `.dvi` or a `.pdf` file.

- If the file is compiled with the `latex` command, it usually produces a `.dvi` file.
- If the file is compiled with the `pdflatex` command, it usually produces a `.pdf` file.
- Several auxiliary files are also created.

Many LaTeX editors can produce both `.dvi` and `.pdf` files.

Special characters

The following symbols are reserved characters that have a special meaning in LaTeX and, when entered directly in text, will coerce LaTeX to do things you did not intend

$ & % # _ { } ~ ^ \

(for example % comments out the line following it)

You can still produce these characters in the text:

- for $, &, %, #, _, { and } type a backslash \ in front of them (for example type \$ to produce $);
- for ~, ^ and \ you need to use special *commands*.

LaTeX commands

LaTeX commands are case-sensitive and consist of a backslash \ followed by

- a string of letters, or
- exactly one non-letter (e.g. a special character).

Commands may have

- no arguments,
- mandatory arguments, which are input in braces { },
- optional arguments, which are input in square brackets [ ].

For example,

```
\framebox[3in]{\textit{Various fonts} are supported by \LaTeX.}
```

produces

Various fonts are supported by LaTeX.

Macros

You can define your own commands using the following syntax:

```
\newcommand{name}[num]{definition}
```

For example, in order to produce $\frac{\partial L}{\partial x}$ we need to type

```
\frac{\partial L}{\partial x}
```

whereas if we put the following line in the preamble

```
\newcommand{\pd}[2]{\frac{\partial #1}{\partial #2}}
```

it is enough to type

```
\pd{L}{x}
```

If the command `name` is already defined, use `\renewcommand`.

E.g. `\renewcommand{\leq}{\leqslant}` replaces $\leq$ with $\leqslant$.

Environments

Environments are building blocks of a LaTeX file. Each declaration of an environment has the following syntax

```
\begin{environment} text \end{environment}
```

Examples include

- `\begin{document} ... \end{document}`,
- `\begin{enumerate} ... \end{enumerate}`,
- `\begin{displaymath} ... \end{displaymath}`.

Environments can be nested within each other so long as the correct nesting order is maintained:

```
\begin{aaa}
    ...
    \begin{bbb}
        ...
    \end{bbb}
    ...
\end{aaa}
```

Typesetting Mathematics

Mathematical text within a paragraph is entered between `$` and `$`, e.g. `$\alpha=\sum_{i=1}^n \beta^i$` produces $\alpha = \sum_{i=1}^n \beta^i$.

To insert an equation on a separate line, write `\[ ... \]` or `\begin{displaymath} ... \end{displaymath}`, e.g.

```
\[ \alpha = \sum_{i=1}^n \beta^i \]
```

produces

$$\alpha = \sum_{i=1}^{n} \beta^{i}.$$

To produce in line formulae with `\displaymath` layout, use `$\displaystyle...$` instead of `$...$`.

To number your equation, use the `equation` environment.

To vertically align equations, use the `align` or `align*` environment.

Including graphics

There are several ways of including graphics in LaTeX and you will most likely need the `graphicx` and `color` packages.

`Creating graphics in LaTeX` is possible, but might be quite time-consuming and you cannot create complex pictures.

Including graphics

However, you can draw a picture elsewhere and then insert it in LaTeX using special commands!

`To include EPS graphics`, e.g. *file.eps* type

```
\begin{figure}[ht]
  \centering
  \includegraphics[options]{file.eps}
  \caption{My figure}
  \label{the-label-for-cross-referencing}
\end{figure}
```

Note that you cannot insert an `.eps` file into a `.pdf` file.

`To include PDF/PNG/JPEG graphics`, use the same algorithm but only if LaTeX generates a `.pdf` file.

Including graphics

Using the program called xfig you can draw figures and write LaTeX on them. Insert such figures into LaTeX using the following commands:

```
\begin{figure}[ht]
  \centering
  \input{file.pstex_t}
  \caption{My figure}
  \label{the-label-for-cross-referencing}
\end{figure}
```

BibTeX

BibTeX is a tool that generates a list of references from a bibliographical database.

- You maintain one file in which you contain information about all possible articles you may wish to reference.
- You only specify the style and location of the bibliography.
- No need to retype the same references for your next article.

A typical entry in a `.bib` file looks as follows:

```
@article {GM,
    AUTHOR={Gowers, W. T. and Maurey, B.},
     TITLE={Banach spaces with small spaces of operators},
   JOURNAL={Math. Ann.},
      YEAR={1997},
    NUMBER={4}
     PAGES={543--568}}
```

Often you are able to copy-paste this information directly from MathSciNet — go to "Select alternative format" and select BibTeX.

All this was just the tip of the iceberg. With LaTeX you can do *much* more:

- create title pages, fancy headers and footnotes;
- split text into chapters, sections, . . . , and create table of contents;
- create enumerated and bullet point lists;
- create theorems and proofs;
- label and cross-reference;
- manipulate counters;
- insert hyperlinks;
- write text in different languages;
- . . .

In order to have LaTeX on your home PC, you will need to install

- TeX distribution
  - MiKTeX (Windows), TeX Live (most common operating systems), proTeXt (Windows), MacTeX (Mac OS X), ...
- LaTeX editor (although you can write LaTeX even in notepad)
  - Texmaker (Windows, Unix, Mac OS X), LEd (Windows), kile (Unix/Linux X-windows systems, Mac OS X), TeXnikCenter (Windows), Emacs (most systems), Google Docs (LaTeX lab), ...
- program(s) for viewing output LaTeX files
  - YAP, Adobe Reader, ...
- possibly something else
  - GhostScript, ...

Watch compatibility between various, particularly the newest, versions of software.

> You know we all became mathematicians for the same reason: we were lazy.
>
> Max Rosenlicht

Be lazy! Let LaTeX do as much of your work as possible. Use

- labels and cross-references,
- sectioning commands and theorem environments,
- enumeration environments,
- tabular, array and align environments,
- macros,
- BibTeX.

To avoid errors:

- compile the document as often as possible;
- having written `\begin{xxx}`, add `\end{xxx}` and only then typeset the middle part; same for { and }.

- Distinguish between *italic* and roman fonts in math mode. Compare

$$\int_0^1 e^{inx} cosnxdx = 1 \quad for\ n = 0$$

and

$$\int_0^1 \mathrm{e}^{inx} \cos nx \mathrm{d}x = 1 \quad \mathrm{for}\ n = 0.$$

Use roman alphabet for

- non-mathematical symbols,
- differential `d`, exponential `e`, complex `i` and other reserved letters,
- functions like `sin`, `cos`, `log` etc. (for these you need to use the backslash version, i.e. write `\cos` instead of `cos`).

- Add punctuation after equations and inside enumeration environments.
- Differentiate between

-	hyphen	X-ray,
–	en-dash	pages 1–12, Cauchy–Schwartz inequality, and
—	em-dash	a punctuation dash — like this.

  How many authors does the Birch–Swinnerton-Dyer conjecture have?
- Do not use " for quotation marks. Instead type

two `	(grave accent)	for opening quotation marks, and
two ′	(vertical quote)	for closing quotation marks.

- Don't be frustrated if something doesn't work out — Google is always there to help you!
  (just don't google the word "latex" on its own . . .)

- This talk, **The Not So Short Introduction to LaTeX 2_ε** on which the talk is based, plus a lot of other information is available at http://www.maths.ox.ac.uk/help/faqs/latex/;
- MiKTeX: http://miktex.org/;
- TeX Live: http://www.tug.org/texlive;
- Texmaker: http://www.xm1math.net/texmaker/;
- LEd: http://www.latexeditor.org;
- Kile: http://kile.sourceforge.net;
- The Comprehensive TeX Archive Network (CTAN): http://www.ctan.org;
- Detexify: http://detexify.kirelabs.org/classify.html;
- Google.