LaTeX Workshop

Roman V. Belavkin

December 8, 2009

Contents

1 Introduction to LaTeX

Brief history of TeX and LaTeX

- Donald Knuth first thought of creating TeX in 1977 after he received the print proofs of his book from the publisher.
- The first versions of TeX appeared in the beginning of 1980s (Knuth, 1984)
- Because TeX is essentially a programming language, many extra packages appeared (e.g. AMSTeX)
- LaTeX is an 'extra' package (higher level language) for TeX that was created in the 1980s by Leslie Lamport, and it soon became the most popular and standard way of using TeX.

Why LaTeX?

- It was created by scientists (academics) for scientists, and it is *de facto* the standard system used by the majority of research community (publishers, conference organisers, societies)
- It has outstanding typographic quality not matched by any other commercial or non–commercial system with the same level of features.
- It is not an editor, but a document markup language (programmable), much like `html`, and so your work does not depend on the text editor you use.
- Your work is saved as text documents (`*.tex`), which means its is very difficult to loose (unless you don't backup).
- It is free.

Why LaTeX?

- It has been specifically designed to handle automatic enumeration of objects, typeset complex objects (e.g. math formulae), use non–standard fonts, include graphics and bibliography.

- LaTeX philosophy was to allow authors concentrate on the contents, not on the layout. The layout is handled by style templates automatically (`css` borrowed the idea), and they are easy to change.
- You have complete control of your work.

Overview of the main features

- LaTeX is a programming language with rich set of commands.
- It has the most complete set of mathematical fonts, objects and symbols.
- LaTeX makes most of the formatting decisions for you, and if you want, you can change anything the way you like.
- It automatically enumerates objects (sections, equations, tables, figures, etc), and you don't need to worry about enumeration if you decide to change the order.
- There are large libraries of additional packages to create complex objects (e.g. large tables, graphics, diagrams, conditional execution, etc).
- LaTeX formats and produces documents of the highest typographic quality and using many formats (PS, PDF, etc).

More on quality

TeX and LaTeX formatting algorithms include features not found in any standard word processors, and only few professional typesetting systems can match the quality.

- Kerning (adjusting the spacing between letters)
- High quality fonts including real small caps, common ligatures, rare and ancient ligatures
- Glyph variants
- Line breaks, justification and hyphenation
- Microtypography

See `http://nitens.org/taraborelli/latex`

Installation

- Most common distributions are TEXLive

  `http://www.tug.org/texlive/`

  and MiKTEX

  `http://www.miktex.org/`

  that include LATEX as well as other systems.

- It is a standard package on Linux, but it can be installed on most platforms (e.g. Windows, MacOS)

- On MS Windows you can install Cygwin (Linux for Windows), which also includes TEXLive

How it works

1. Create and type your document as a standard textfile (usually `*.tex`) in any text editor (e.g. Notepad, Emacs)

   `file.tex`

2. Save and compile it by executing

   `latex file.tex`

3. This will create the output file `file.dvi`, which you can preview, print, convert to PDF.

4. Sometimes you need to compile twice for the automatic re-enumeration to work

Document structure

first.tex

```
\documentclass[12pt]{article}

\title{My first document}
\author{First Last}
\date{\today}

\begin{document}
\maketitle

\section{About today}
Today I am learning \LaTeX.

\end{document}
```

Save the file as `first.tex`

Compiling and previewing

- Type on the command prompt

  `latex first.tex`

  Alternatively, select `TeX -> TeX File` in Emacs (`Ctrl-c-f`).

- To preview type the command

  `xdvi first.dvi&`

  Alternatively, select `TeX -> TeX View` in Emacs (`Ctrl-c-v`).

- You can keep Xdvi open to view the changes you make.

Producing PDF

- One way is simply to use PdfLaTeX:

  `pdflatex first.tex`

- Another way is to first create PS from your `*.dvi` file, and then PDF

```
dvips   -P pdf first.dvi
ps2pdf  first.ps
```

Separation of style from content

- When typing in LaTeX, you are encouraged to think about what you write, not how it looks. You can change the way it looks later.
- You can, if you want, adjust everything (e.g. put a forced pagebreak)
- It is easy to change the whole style. Fro example, try changing the style from

  ```
  \documentclass[12pt]{article}
  ```

  to

  ```
  \documentclass[12pt]{report}
  ```

- Or use two columns

  ```
  \documentclass[12pt,twocolumn]{report}
  ```

2 Basics of typesetting in LaTeX

Paragraphs

```
Text is just text, where words are
separated by spaces.  It does not matter
how many spaces I type, because
\LaTeX\ will take care of formatting.
If I want to start a new paragraph, then
I simply type from a new line.

So, this sentence is already a new paragraph.
Usually, it is not a good idea to have one
sentence paragraphs.
```

Special characters

`\` backslash tells LaTeX that you are starting a command (`\box`)

`%` starts a comment.

`$` starts or ends math mode (equations).

`_` underscore is used in math mode for subscript X_i

`^` hat is used in math mode for superscipt X^2

`&` to tabulate table columns.

`~` tilde is used to insert unbreakable space

`#` hash is used in definition of arguments

`{}` curly brackets are used in names of commands

If you need these signs, simply use commands:

```
$\backslash$ \% \$ \_ \^ \& \~ \# \{ \}
```

Symbols and fonts

Example

```
In Britain we use \pounds, but americans use \$.
There are many other symbols, such as \copyright, and
many fonts like {\bf bold face} or {\em italics}.

Mathematicians like to use Greek letters,
such as $\alpha$, $\beta$, $\gamma$ and
even capitals like $\Gamma$.
```

In Britain we use £, but Americans use $. There are many other symbols, such as ©, and many fonts like **bold face** or *italics*.

Mathematicians like to use Greek letters, such as α, β, γ and even capitals like Γ.

Types of LaTeX commands

- All commands begin with `\`
- Curly brackets are used for scoping

  ```
  {\bf This text is bold} and this is normal.
  ```

- Simple commands or symbols have no arguments

  ```
  \tableofcontents  \alpha \beta \rightarrow
  ```

- Some commands are followed by arguments `{}` and options `[]`

  ```
  \section{Inroduction}
  \usepackage[all]{xy}
  ```

- Environments have `\begin{..}  \end{..}` form

  ```
  \begin{equation}
  E = m c^2
  \end{equation}
  ```

Mathematical equations

Example

```
Equations can be very short,
but very important, like
this by Einstein
\begin{equation}
E = m c^2
\end{equation}
And some are believed to be
beautiful, like this
by Euler
\[
e^{i\pi} + 1 = 0
\]
```

Equations can be very short, but very important, like this by Einstein

$$E = mc^2 \tag{1}$$

And some are believed to be very beautiful, like this by Euler

$$e^{i\pi} + 1 = 0$$

In-text and Display math modes

Example

```
There are two kinds of math mode.  If I need to type
a short formula in text, I use in-text math mode,
like this $E\{x\}=\sum X_i P_i$.  If the formula is
long and important, I use the display math mode,
like this
\[
\sum_{i=1}^m X_i P_i = X_1 P_1 + \cdots + X_m P_m
\]
Some displayed equations are enumerated
\begin{equation}
E\{x\} = \int_A x(a) P(da)
\end{equation}
```

There are two kinds of math mode. If I need to type a short formula in text, I use in-text math mode, like this $E\{x\} = \sum X_i P_i$. If the formula is long and important, I use the display math mode, like this

$$\sum_{i=1}^{m} X_i P_i = X_1 P_1 + \cdots + X_m P_m$$

Some displayed equations are enumerated

$$E\{x\} = \int_A x(a) P(da) \tag{2}$$

Document structure: Sections, subsections

- Let us add some structure by adding

  ```
  \subsection{Commands}
  ```

  before the paragraph '*In Britain we use £*', and

  ```
  \subsection{Mathematical formulae}
  ```

  before the paragraph '*There are two kinds...*'

- Start a new section in the end of the text

  ```
  \section{Labelling and referencing objects}
  ```

Labelling and referencing objects

- Type the following text in the new section

  ```
  Numbers are important, but they are easy
  to forget. Their order can be confusing.
  In \LaTeX\ we give objects meaningful labels.
  ```

- Find the Einstein's equation, and add the following label

```
\begin{equation}
E = m c^2   \label{Einstein}
\end{equation}
```

- Now, continue typing in the new section.

```
Einstein's formula is very short, but I don't
remember that its number is~(\ref{Einstein}).
```

- Compile your document twice and preview.

Changing the order is easy

- Now cut the block starting from '*Equations can be very short...*' and ending by the Euler's formula
- Paste this block in the end of your first section (before section '*Labelling and referencing objects*'
- Compile the file, and notice:

`LaTeX Warning: Label(s) may have changed...`

- Compile again and see what's happened

Changing preamble

- Go into the beginning of your document, and add the following command just after `\maketitle`:

```
\tableofcontents
```

- Compile twice and see the effect
- In a similar way, LaTeX can create index, author index, list of figures, tables and so on.
- In the next tutorial we shall learn about handling bibliography, citations and references.

Matrices

Example

```
Matrices are mathematical arrays that look like this
\[
\left(
\begin{array}{cc}
2 & 0 \\
0 & 2
\end{array}
\right)
\]
```

Matrices are mathematical arrays that look like this

$$\left(\begin{array}{cc} 2 & 0 \\ 0 & 2 \end{array} \right)$$

Other mathematical arrays

Example

```
Some arrays do not have two brackets
\[
\delta_a(b)=
\left\{
\begin{array}{cc}
1 & \mbox{if $a=b$}\\
0 & \mbox{otherwise}
\end{array}
\right.
\]
```

Some arrays do not have two brackets

$$\delta_a(b) = \left\{ \begin{array}{cc} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{array} \right.$$

Tables

Example

```
Tables may contain mostly text or just numbers
\begin{tabular}{|r|lll|}
\hline
No. & Name & Odd & Prime \\
\hline
1 & One   & Yes & Yes \\
2 & Two   & No  & Yes \\
3 & Three & Yes & Yes \\
4 & Four  & No  & No  \\
$\vdots$& &     &     \\
m & & & \\
\hline
\end{tabular}
```

Tables may contain mostly text or just numbers

No.	Name	Odd	Prime
1	One	Yes	Yes
2	Two	No	Yes
3	Three	Yes	Yes
4	Four	No	No
⋮			
m			

Table captions and numbers

Put your table in the following environment, add caption and label.

Example

```
\begin{table}[ht]
\caption{Classification of numbers}
\label{Numbers}
\centering
\begin{tabular}{|r|lll|}
...
\end{tabular}
\end{table}
```

Table 1: Classification of numbers

No.	Name	Odd	Prime
1	One	Yes	Yes
2	Two	No	Yes
3	Three	Yes	Yes
4	Four	No	No
⋮			
m			

3 BibTEX: Manage bibliography, citations and references

BibTEX overview

- BibTEXis program for automatic compiling lists of references in your LATEX documents.
- It was created by Oren Patashnik and Leslie Lamport in 1985.
- It can create lists of references in different styles, and you do not have to worry about the formatting.
- Most of the conferences or journals supply their BibTEX styles.
- It only takes editing one command to change the style in your article.
- You can create and keep your bibliography in separate files to re-use in different articles.

How it works

1. Create and type your bibliography as a standard textfile (usually `*.bib`) using any text editor (e.g. Notepad, Emacs)

   `file.bib`

2. In this file, you give each entry a key, such as `Shannon48`
3. If you need to cite this document in your paper, you simply add a command `\cite{Shannon48}`
4. Define bibliography style anywhere in your document

   ```
   \bibliographystyle{plain}
   ```

5. In the end of your document, link your bibliography file by

   ```
   \bibliography{file}
   ```

6. Finally, you need to compile, run BibTEX and then compile again twice

Creating your bibliography file

Let us create the file `first.bib` and type

Book

```
@Book{Picard97,
  author =    {R. W. Picard},
  title =     {Affective computing},
  publisher = {MIT Press},
  year =      1997,
  address =   {Cambridge, MA}
}
```

Journal article

```
@Article{Shannon48,
  author =  {Claude E. Shannon},
  title =   {A mathematical theory of communication},
  journal = {Bell System Technical Journal},
  year =    1948,
  volume =  27,
  pages =   {379--423 and 623--656},
  month =   {July and October}
}
```

Conference article

```
@InProceedings{Belavkin07:_ijcnn07,
  author =    {Roman V. Belavkin},
  title =     {Do neural models scale up
               to a human brain?},
  booktitle = {{I}nternational {J}oint {C}onference
               on {N}eural {N}etworks ({IJCNN} 2007)},
  year =       2007,
  publisher = {IEEE},
  month =     {August}
}
```

Adding citations

Go back to your document and add the following text

Citations

```
Some computer scientists predict that computers
will have emotions~\cite{Picard97}.
Information theory may provide some answers
to this probelm~\cite{Shannon48,Belavkin07:_ijcnn07}.
```

Linking bibliography to your document

- Add the following commands before the `\end{document}`

  ```
  \bibliographystyle{plain}
  \bibliography{first}
  ```

- Compile, BibTEX and compile twice your document

  ```
  latex first.tex
  bibtex first.tex
  latex first.tex
  latex first.tex
  ```

  Alternatively, select `TeX -> TeX File`, followed by `BibTeX File`, and `TeX File` twice

- Preview `TeX -> TeX View`

Changing bibliography style

- We've used numerical style (plain) in the previous example.
- An important family of bibliography styles is Harvard and particularly its American Psychological Association (APA) version.
- Add the following package in the preamble of your document

  ```
  \usepackage{apacite}
  ```

- Change your bibliography style to

  ```
  \bibliographystyle{apacite}
  ```

- Compile, BibTEX and compile twice.

4 Advanced features: Graphics, Diagrams

Professional fonts

- For more symbols use

  ```
  \usepackage{amssymb}
  ```

- For additional fonts

  ```
  \usepackage{amsfonts}
  ```

Adding graphics

- To use graphi files, use

  ```
  \usepackage{graphicx}
  ```

- To add graphic file `picture.eps` use the command

  ```
  \includegraphics{picture.eps}
  ```

- Or better even

  ```
  \begin{figure}[ht]
  \centering
  \includegraphics{picture.eps}
  \caption{My figure}
  \label{myfigure}
  \end{figure}
  ```

Using GNUplot

- You can use GNUplot to create various graphis files

Diagrams

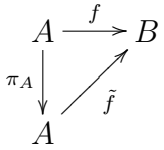
- To draw diagrams you can use

```
\usepackage[all]{xy}
```

- Try this

```
\[
\xymatrix{ A \ar[r]^{f}
  \ar[d]_{\pi_A} & B\\
A \ar[ru]_{\tilde{f}} &}
\]
```

- Will draw this



If-Then

- For conditional execution

```
\usepackage{ifthen}
```

- Then you can define and initialise Boolean vatiable:

```
\setboolean{printsolution}{false}
```

- And use the `\ifthenelse{}{}{}` command:

```
\ifthenelse{\boolean{printsolution}}
{This text is printed when the variable is true}
{This text is printed otherwise}
```

Other resources

- One of the main resource is TEX users' group:

`www.tug.com`

5 Preparing your conference paper in LaTeX

6 Using LaTeX for teaching

References

Knuth, D. E. (1984). *The texbook*. Addison-Wesley.

Lamport, L. (1994). *Latex: A document preparation system: User's guide and reference*. Addison-Wesley.