I ♡ LaTeX

by Marina Bykova

February 22, 2001

1 What is LaTeX?

TeX is a powerful text processing language which was developed to make your documents look beautiful. TeX is a *text processor* rather than a *word processor* and it is not WYSIWYG (What You See Is What You Get) like many of the available word processors. It is arguably the premier typesetting package in the world. Knuth and Lamport have distilled for us the accumulated wisdom of generations of printers.

LaTeX is a macro package which sits on top of TeX and provides all the structuring facilities to help with writing large documents. LaTeX has the reputation of being hard, but in fact it is effectively the same as HTML!

Why LaTeX?

1. It is free on virtually every computer in the world;
2. LaTeX source files are purely alphanumeric so it can be read by eye or posted by e-mail with no problems associated with different versions or binary files;
3. You can use *jed* to write your LaTeX programs;
4. The typesetting is better, it is simply the best package for documents containing mathematics;
5. Style changes are neater in LaTeX. Style files for many periodicals exist.
6. LaTeX is extensible. If you want a new feature, you can look around for a free add-on or write one yourself.

2 What can it do?

LaTeX is cool

Better ask what it can *not* do. Here is an abbreviated list of things that LaTeX includes:

- Great variety of fonts, page formats, and pagenumbering styles;
- Multi-column printing;
- Plenty of special characters, foreign letters, accents, symbols, etc.;
- Footnotes[1] and marginal notes;

marginal notes look like this

[1] No kidding

- Control over line and page breaks, control over word division and hyphenation;
- Cross–references;
- Tables, boxes, and mini-pages;
- Possibility to build new commands, use your own counters, redefine *everything*;
- Different document styles and classes that can be greatly expanded by usage of various packages.

LaTeX is automated

LaTeX not only allows us to write what we want and how we want but it also aids us in document preparation. LaTeX can automatically generate:

- Titles (see the previous page) and title pages;
- Tables of contents;
- Lists of figures;
- Lists of tables;
- Lists of equations;
- Bibliography which is automatically built from your BibTeX database file;
- Keyword indecies;
- Glossaries.

LaTeX is well suited for math

As mentioned previously, LaTeX is the best tool when you need to produce documents containing sophisticated mathematical formulas. It handles and makes statements and formulas look beatiful, starting from simple as $y = 2x^2 + 3$ to

$$\Gamma(x) \equiv \lim_{n\to\infty} \prod_{\nu=0}^{n-1} \frac{n!n^{x-1}}{x+v} \equiv \int_0^\infty e^{-t} t^{x-1} \mathrm{d}t$$

and much more.

Here are a couple of other examples that confirm the stated above:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \Rightarrow U_M = \frac{1}{4\pi} \oint\limits_{\Sigma} \frac{1}{r} \frac{\partial U}{\partial n} \mathrm{d}s - \frac{1}{4\pi} \oint\limits_{\Sigma} \frac{\partial \frac{1}{r}}{\partial n} U \mathrm{d}s$$

$$\sum_{p_1<p_2<\cdots<p_{n-k}}^{(1,2,\ldots,n)} \Delta \begin{matrix} p_1 p_2 \cdots p_{n-k} \\ p_1 p_2 \cdots p_{n-k} \end{matrix} \sum_{q_1<q_2<\cdots q_k} \begin{vmatrix} a_{q_1 q_1} & a_{q_1 q_2} & \cdots & a_{a_1 q_k} \\ a_{q_2 q_1} & a_{q_2 q_2} & \cdots & a_{a_2 q_k} \\ \multicolumn{4}{c}{\dotfill} \\ a_{q_k q_1} & a_{q_k q_2} & \cdots & a_{a_k q_k} \end{vmatrix}$$

$$e^x = 1 + \frac{x}{1 - \frac{x}{2 + \frac{x}{3 - \frac{x}{2 + \frac{x}{5 - \cdots}}}}}$$

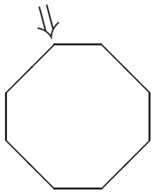
Do you like it?

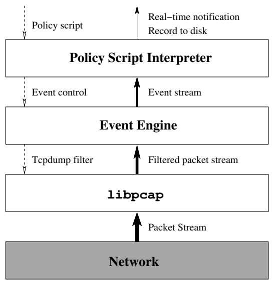
where
I can write
I want and
the way I want.

Graphics in LaTeX rules

Diagrams, photographs, screen shots, plots or experimental data, etc. can be included in LaTeX files. However, simple datagrams, pictures, and plots can be also produces by means of the language itself. Here are some simple examples.

by means of LaTeX





⟵ by means of `xfig`

You see how cool we can be if we do what we like, and especially if we have time

3 Getting Started

The "Hello, World!" Program

1. Start your favorite editor, i.e. jed.
2. Type in file *hello.tex*

```
\documentclass{article}
\begin{document}
% this is my first beautiful paper
Hello, World!
\end{document}
```

3. Compile your program

```
latex hello
dvips -o hello.ps hello
```

The first command generates a `.dvi` (*device–independent*) file, or *metafile*, that contains the *formatted* text along with information about the required character fonts, but in a form that is independent of the characteristics of the printer to be used. The second command converts the `.dvi` file to a PostScript `.ps` file.

In some cases you will need to run the first command more than once. It is needed when you use cross-references (to reference books, chapters, sections, tables, etc.). On the first run, LaTeX builds a database of references, and it includes correct values in the document itself on the second run.

4. View your paper

```
gv hello &
```

4 Document Layout and Organization

4.1 Structure of a LaTeX document

In every LaTeX file there must be a *preamble* and a *body*.

The preamble is a collection of commands that specify the global processing parameters for the folowing text, such as the paper format, the height and width of the text, the form of the output page with its pagination, etc. At the very least, the preamble must contain the command `\documentclass` to specify the document overall processing type.

The standard document classes are *book*, *article*, *report*, and *letter*. They can be found in `.cls` files. You can specify your own class after you have written a new `.cls` or found it somewhere else.

The preamble ends with `\begin{document}`, and everything that follows this command is interpreted as body with actual text. The general syntax of a LaTeX file is as follows:

`\documentclass[`*options*`]{`*class*`}`
Global commands and specifications
`\begin{document}`
Text mixed with additional commands
`\end{document}`

For example, you can use:

```
\documentclass[12pt,titlepage,twocolumn]{article}
```

4.2 Packages

Packages allow us to change the way certain commands behave and define new commands to add extra features that are not part of the standard LaTeX. A package is nothing more than a set of LaTeX commands stored in a file with the extention `.sty`. To invoke a package, simply call

`\usepackage[`*options*`]{`*packages*`}`

in the preamble where *options* are optional. For example, you might use

```
\usepackage[dvips]{graphics,color}
```

Description of a few packages that can be useful is included below.

ifthen — provides conditional commands `\ifthenelse{`*test*`}{`*then_test*`}{`*else_test*`}` and `\whiledo{`*test*`}{`*do_test*`}` where you can test for numbers, text, lengths, or switches.

doublespace — provides double spacing which can be required for certain types of documents

graphics — needless to say, this package adds additional features and commands to the standard LaTeX that allow us to manipulate graphics easier and create more interesting solutions. The package was used to create this paper as well.

geometry — a useful package that makes it easier to configure many settings of your document, for instance margins, without direct manipulation with those lengths.

4.3 Special Characters

The *space* or *blank* character has some properties different from those of normal characters. During processing, blanks in the input text are replaced by rubber lengths in order to allow the line to fill up to the full line width. As a result, the following rules apply:

- one blank is the same as a thousand;

- blanks at the beginning and end of an input line and blanks terminating a command name are ignored;
- *carriage returns* are treated as blanks;
- two carriage returns in a row produce a new paragraph.

In book printing, the character `-` comes in various length: -, –, —. Such characters can be generated by tyoing the hypen character one. two, or three times, so that `-` yelds -, while `--` makes –, and `---` produces —. A forth type of dash is the munis sign $-$, which is entered in math mode as `$-$`.

The single characters `# $ & ~ _ ^ % { }` all have special meanings and are interpreted as commands. To print them as text, you must give a command consisting of \ plus that character.

$ = `\$`   & = `\&`   ˜ = `\~`   _ = `\_`   ˆ = `\^`   % = `\%`   { = `\{`   } = `\}`

There are many more special characters, foreign letters, accents, and symbols like ö, ©, $\pm$, $\Uparrow$, $\hbar$, and $\leadsto$ available but you will explore them on your own.

4.4 Lenghts

Lengths consist of a decimanl number with a possble sign in front (`+` or `-`) followed by a mandaroty dimensional unit. Some of the permissible units are:

`cm`	centimeter,
`mm`	millimeter,
`in`	inch (1 in = 2.54 cm),
`pt`	point (1 in = 72.27 pt),
`pc`	pica (1 pc = 12 pt).

Values are assigned to a length parameter by means of the command `\setlength`. Its syntax is as following:

`\setlength{`*\length_command*`}{`*length_spec*`}`

For example, you can use:

```
\setlength{\textwidth}{12.5cm}
\setlength{\topmargin}{1in}
\setlength{\oddsidemargin}{2,5cm}
```

4.5 Sections

The following commands are available for producing automatic sectioning:

`\section`   `\subsection`   `\subsubsection`

The syntax of these commands is:

`\section{`*section_name*`}`	creates a numbered section
`\section*{`*another_section*`}`	creates an unnumbered section

In the first case, the section is given the next number in the sequence, which is printed together with a heading using the text *section_name*. In the second case, no section number is printed and no entry in the table of contents is made.

4.6 Maketitle

If you want to generate a title for your article (or a title page for your book), you need to copy the following commands into your document:

```
\title{My Sexy \LaTeX\ Paper}
\author{Scott Barron}
\date{February 29, 2001} ←— optional
```

If you omit date declaration, LaTeX will use the current date. The value of the current date can also be obtained using the command `\today`[2].

Command `\maketitle` will generate a title for you. You need to make sure that the commands appears *after* you specified the title, authors, and date and *before* any text in the body of your document.

5 Displayed Text

5.1 Simple Font Manipulations

5.1.1 Font Types

In case you want to change a font you are using, there are a number of *font commands* that set their argument in a font with the specified attribute. Some of the commands are:

`\textrm`{*text*}	Roman	`\texttt`{*text*}	`Typewriter`
`\textsf`{*text*}	Serif	`\textit`{*text*}	*Italic*
`\textsl`{*text*}	*Slanted*	`\textsc`{*text*}	SMALL CAPS
`\textbf`{*text*}	**Bold font**	`\textnormal`{*text*}	Default

5.1.2 Font Size

The three possible basic font sizes are 10, 11, and 12pt, depending on the size options `10pt` (default), `11pt`, and `12pt` specified in the document class. The following declarations are available in LaTeX for changing the font size:

`\tiny`	smallest	`\Large`	larger
`\scriptsize`	very small	`\LARGE`	even larger
`\footnotesize`	smaller	`\huge`	still larger
`\small`	small	`\Huge`	largest
`\normalsize`	normal		
`\large`	large		

all of which are relative to the standard size selected in the document class option.

5.2 Spacing

Spacing of any desired size may be inserted into the text with the command

`\hspace`{*space*}

Vertical spacing between particular paragraphs can be increased or decreased using the command

`\vspace`{*space*}

[2]Other commands `\day`, `\month`, and `\year` return the current values of theses parameters as numbers.

The commands

`\bigskip` `\medskip` `\smallskip`

can also be used to increase the spacing between paragraphs.

To redefine spacing between paragraphs and paragraph indentation, you can use the following commands:

`\setlength{\parskip}{`*space*`}`
`\setlength{\parindent}{`*space*`}`

To suppress indentation for one paragraph or force it where it would otherwise not occur, place

`\noindent` or `\indent`

Another useful LaTeX command is `\\`. It allows us to achieve a new line without going to a new paragraph.

5.3 Centering

The environment

`\begin{center}` *line 1* `\\` *line 2* `\\` ... *line n* `\end{center}`

centers the sections of the text that are separated by the `\\` command. A single line may centered by typing its text as the argument of the TeX command `\centerline{`*text*`}`

5.4 Lists

There are tree environments available for producing formatted lists:

`\begin{itemize}`	*list text*	`\end{itemize}`
`\begin{enumerate}`	*list text*	`\end{enumeate}`
`\begin{description}`	*list text*	`\end{description}`

The environment `itemize` gives us lists where individual entries are indicated with bullets, in the environment `enumerate` they are numbered, and in the environment `description` they appear as key words and their definitions. The command to produce the labels in the list text is `\item` or `\item[`*option*`]` for the environment `description`.

The above lists may be included within one anotherm eitehr mixed or of one type, to a depth of four levels.

Example

You type the following text in your research paper:

```
What I learned today is:
\begin{enumerate}
  \item Some basic definitions:
   \begin{description}
    \item[jed] A superior editor of all times and nations.
    \item[latex] The only program that will make my documents look beautiful.
   \end{description}
  \item The first commands that I type on my computer today when I come home will
  be \texttt{`xjed latex\_my\_love.tex \&`}.
\end{enumerate}
```

In the generated document, it will appear as:

What I learned today is:

1. Some basic definitions:

   jed A superior editor of all times and nations.

   latex The only program that will make my documents look beautiful.

2. The first commands that I type on my computer today when I come home will be `‘xjed latex_my_love.tex &‘`.

5.5 Mathematical Formulas

Mathematics is the soul of TEX. Actually, Donald Knuth invented his text formatting system because the setting of mathematical formulas is do complicated in normal printing.

The processing of *math text* is carried out by switching to *math mode*. There are two different types of how math text may occur in a document which are referred as *text formulas*, for example $(a+b)^2 = a^2 + 2ab + b^2$, and *displayed formulas*, for example

$$(a+b)^2 = a^2 + 2ab + b^2$$

Text formulas, or equations, are generated with the environment

`\begin{math}` *formula_text* `\end{math}`

A shorthand version is available as `$`*formula_text*`$`.
Displayed formulas, or equations, are produced in the environments

`\begin{displaymath}` *formula_text* `\end{displaymath}`
`\begin{equation}` *formula_text* `\end{equation}`

The difference between these two is that the `equation` envornment automatically adds a swqueatila equation number. The `displaymath` environment may be given with the shorthand form `$$`*formula_text*`$$`.

It is worth mentioning that some characters can be displayed only in math mode. Do not get confused if you see “Missing $” error during compilation. Examples of such characters include

$|$ = `$|$` $\backslash$ = `$\backslash$` x_n = `x$_n$`

And of course there are a huge variety of mathematical symbols, letters, and signs available for creation of the most sophisticated mathematical formulas.

6 What to do next

Everything else that was not mentioned here can be easily found in LATEX documentation. There a lot of different resources on the web. Some of them are the Latex Project web site [Pro00] and Cambridge University Text Processing web site [UoC00]. Others can be found by typing `latex` at Google.com.

There are a number of good books listed on the Latex Project web site that can aid in writing LATEX documets [Lam94], [GMS94], [GRM97]. Another good book “A Guide to LATEX” by Helmut Kopka [KD99] was heavily used to create this guide.

References

[GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Pub Co, January 1994.

[GRM97] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley Pub Co, April 1997.

[KD99] Helmut Kopka and Patrick Daly. *A Guide to LaTeX*. Pearson Education Ltd, 1999.

[Lam94] Leslie Lamport. *LaTeX: A document Preparation System*. Addison-Wesley Pub Co, April 1994.

[Pro00] The LaTeX 3 Project. LaTeX: A document preparation system, 2000. http://www.latex-project.org/.

[UoC00] Department of Engineering University of Cambridge. Text processing using LaTeX, December 2000. http://www.eng.cam.ac.uk/help/tpl/textprocessing/.