Guide to LaTeX

Daniel M. Kane

Fall 2022

1 Overview

TeXis a typesetting program developed by Donald Knuth. LaTeXwas an extension of TeXcreated by Leslie Lamport, making a number of operations considerably easier. As TeXand LaTeXare both well suited to typesetting mathematical formulae, which can be annoying to do otherwise, they have become somewhat of a standard in the mathematics community (also computer science and physics and probably a few more). Essentially, either system takes .tex files and compiles them into .pdf documents, or one of a few other formats.

LaTeXprovides a convenient way to produce nicely formatted mathematical documents. In addition to being recommended for use in writing up homeworks for CSE 101, LaTeXmay prove useful to many of you later in your careers. Thus, for those of you who do not have previous experience with LaTeX(or just need a refresher) the following guide can hopefully help you get started. For those of you looking for another starting reference, consult Google, or check out the guide on the Art of Problem Solving website [1]. If there is something that you would like to do in LaTeXthat is not covered here, generally a websearch for an appropriate phrase like "latex underline" or "latex large font" will be enough to figure out how to do it.

2 Software

In order to use LaTeX, you will need to download a compiler for it. For this purpose, you can download MiKTeX from their homepage [6] (it's free). You will probably also want a LaTeXeditor. Fortunately, MiKTeX comes with TeXworks, which acts basically like a text editor along with a compile button. There are other more sophisticated editors as well. Many people use `https://www.overleaf.com/` or `https://www.sharelatex.com/`, which are online editors with some added collaboration features. Personally, I use WinEdt [8], which I would recommend if you are willing to pay the $40 for a student copy.

3 Getting Started

Your LaTeXdocument should start with the line:

```
\documentclass{article}
```

There are document classes other than "article" that can be used, and options that can be applied, but we are going to ignore them for the purposes of this guide. You also need the commands

```
\begin{document}
and
\end{document}
```

Any text that you have between the begin and end document commands will show up in the compiled file. Note that the compiler will ignore most white space that you add. Adding more than one space between words will not affect anything. Additionally, single line breaks are ignored, though double line breaks will produce new paragraphs. Thus:

```
\documentclass{article}
\begin{document}
This             line
    has         lots        of
     extra           spaces.

This line is on a new paragraph.
\end{document}
```

Will produce the output:
This line has lots of extra spaces.
This line is on a new paragraph.

4 Math Formatting

4.1 Math Mode

In order to produce mathematics within LaTeX, expressions encased within $'s are in math mode. If you want the equations to show up on their own line, put them between double $'s. So for example:

```
Inline equation: $x=7.$

Displayed equation: $$ x=7. $$
```

produces:
Inline equation: $x = 7.$
Displayed equation:

$$x = 7.$$

4.2 Superscripts and Subscripts

Within math mode it is easy to produce superscripts and subscripts. For superscripts use the ^ character and for subscripts use the _ character. So for example:

```
$$ x^7 + y_i $$
```

produces:

$$x^7 + y_i$$

Generally, the superscript or subscript is simply the next character. If you want to do more, put the desired super/subscript between { }'s. For example:

```
The 10th power of $x$ is $x^{10}$ not $x^10$.
```

produces:
The 10th power of x is x^{10} not x^10.
You can also have characters with both super- and sub-scripts, or have nested super/sub-scripts. For example:

```
$$ x^{y^z_1}_{a+b} $$
```

produces:

$$x^{y^z_1}_{a+b}$$

4.3 Fractions

Fractions are produced with a \frac command. This takes two arguments, the numerator and then the denominator. Again, if either argument is more than one character long, it should be encased in braces. So, for example:

```
$$ \frac{x^7+y^7}{x+y} $$
```

produces

$$\frac{x^7+y^7}{x+y}$$

4.4 Other Commands

LaTeXhas a bunch of other commands to produce various symbols. All of them start with a \ character and are followed by some text. Some of them take additional arguments. Some useful examples are:

```
$$ \cdot, \sum, \int, \alpha, \Delta, \approx, \sin, \ldots, \{,
\leq, \geq, \rightarrow, \sqrt{x}, \mathbb{Z}, \mathbb{R},\infty$$
```

which produce

$$\cdot, \sum, \int, \alpha, \Delta, \approx, \sin, \ldots, \{, \leq, \geq, \rightarrow, \sqrt{x}, \mathbb{Z}, \mathbb{R}, \infty$$

For more symbols check out [2]. Note that all LaTeXcommands are case sensitive, so make sure that you get capitalization correct. It should be noted that some of these symbols, like the summation and integration symbols, cause super/sub-scripts to be placed immediately above/below the symbol instead of to the right. For example,

```
$$\sum_{n=1}^\infty \frac{1}{n^2}$$
```

produces:

$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

4.5 Parentheses

Using `()` or `[]` will give you parentheses of a standard size (braces are special characters, to get them you need `\{ \}`). Unfortunately, standard sized parentheses look funny when containing tall expressions. To fix this, you can instead use `\left( \right)` or `\left[ \right]`, etc. and your parentheses will rescale themselves to whatever they contain. For example:

```
If you do $$(\frac{x^2+1}{y_2})^n$$ it looks silly. Instead,
$$\left(\frac{x^2+1}{y_2}\right)^n$$ looks much better.
```

which makes:
If you do

$$(\frac{x^2+1}{y_2})^n$$

it looks silly. Instead,

$$\left(\frac{x^2+1}{y_2}\right)^n$$

looks much better.

4.6 Typeface

Many of these can also be done outside of math mode. To get different typefaces, use an appropriate command, with the words you want formatted that way in braces. For example:

```
\emph{Emph}, \textbf{Textbf}, \textsc{Textsc}
```

produces:
Emph, **Textbf**, TEXTSC
Some others that can be used in math mode include

```
$$ \mathrm{mathrm}, \mathbf{mathbf}, \mathbb{MATHBB},
\mathcal{MATHCAL}, \mathfrak{mathfrak}, \overline{overline},
\widehat{widehat}, \overrightarrow{ora}$$
```

The above produces:

$$\mathrm{mathrm}, \mathbf{mathbf}, \mathbb{MATHBB}, \mathcal{MATHCAL}, \mathfrak{mathfrak}, \overline{overline}, \widehat{widehat}, \overrightarrow{ora}$$

Note that mathbb and mathcal do strange things to lower case letters. Mathbb is very useful for producing things like $\mathbb{R}$.

4.7 Special Characters

One thing to be wary of is that `{, }, \, $, &, %, _, ^` are all special characters that are treated unusually. In order to actually produce any of these characters in your output, you need to use `\{, \}, \backslash, \$, \&, \%, \_, \^{}` instead.

5 Sectioning

To divide your document into sections use the `\section` command. For example, this document has the command

```
\section{Sectioning}
```

a couple of lines above this. The section numbering is handled automatically (though there are ways to mess with it if you really want to). You can even do subsections, and subsubsections, though subsubsubsections don't work. For example, when I add

```
\subsection{See}
\subsubsection{how}
\subsubsection{sectioning}
\subsubsection{works?}
\subsection{It's}
\subsubsection{like}
\subsubsection{this.}
```

I get:

5.1 See

5.1.1 how

5.1.2 sectioning

5.1.3 works?

5.2 It's

5.2.1 like

5.2.2 this.

6 Titles

To give your document a title you should put the lines

```
\author{<Your name>}
\title{<Document title>}
\date{<the date>}
```

before the `\begin{document}` line. Then afterwards, a `\maketitle` command will produce a title for your document. For example, this document used

```
...
\author{Daniel M. Kane}
\title{CSE 101 Guide to \LaTeX}
\date{Winter 2022}

\begin{document}
\maketitle
...
```

If you remove the `\date` command, LaTeXwill fill that line in with whatever the date is on the day you compile your file.

7 Packages

A lot of useful things in LaTeXrequire the use of external packages. MiKTeX will generally download these for you automatically if they are reasonably standard ones, but you need to tell LaTeX which ones you are using. To do so, put a `\usepackage{}` command near the top of your document with the package names put inbetween the braces, separated by commas. Useful packages include amssymb and amsmath (have a bunch of the standard symbol commands), amsthm (stuff on theorem environments, which we'll discuss later). This document uses

```
\usepackage{amssymb, amsmath,amsthm,verbatim,hyperref,color}
```

8 Environments

There are a bunch of other useful environments to know about

8.1 Comments

Anything on a line after a % character is ignored by the compiler. So, for example,

```
This is not a comment. % This is.
But not this.
```

produces:
This is not a comment. But not this.

8.2 Labelled Equations

To produce a labelled full-line equation, you can do the following:

```
\begin{equation}
\textrm{Look at me!}
\end{equation}
```

which produces:

$$\text{Look at me!} \tag{1}$$

We'll talk more about how to reference these labels later.

8.3 Aligned Equations

To produce multiline equations, it is recommended that you use the align, or align* environments (the former has line numbering and the latter does not). To use these, produce the commands `\begin{align*}` and `\end{align*}`. Between these, you are in math mode. Use a double backslash to denote line breaks. Use a single & character per line. LaTeXwill attempt to make these vertically aligned with each other. For example:

```
\begin{align*}
(x+y)\cdot (x-y) & = x^2 - x\cdot y + y\cdot x + y^2\\
& = x^2 - x\cdot y + x\cdot y + y^2\\
& = x^2 - y^2.
\end{align*}
```

produces:

$$\begin{aligned}(x+y)\cdot(x-y) &= x^2 - x\cdot y + y\cdot x + y^2\\ &= x^2 - x\cdot y + x\cdot y + y^2\\ &= x^2 - y^2.\end{aligned}$$

8.4 Cases

This can be used to define things by case analysis. For example:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

Such things are produced by putting `\begin{cases},\end{cases}` statements in math mode. As in an align environment, you use double backslash for line breaks and & for alignment. For example, the above was produced by

```
$$
|x| = \begin{cases} x & \textrm{if } x\geq 0 \\
 -x & \textrm{if } x<0 \end{cases}
$$
```

8.5 Theorem Environments

Theorem environments allow you to cleanly produce and reference statements of theorems in your document. In order to use them, you want to use the amsthm package. You also need to use some `newtheorem` commands at the top of your document, before the `\begin{document}`. The general syntax is as follows:

```
\newtheorem{<environment name>}{<caption>}
```

The environment name is the one used to create that type of theorem environment, and the caption is what is displayed at the top when you compile the environment. For example, this document has:

```
\newtheorem{thm}{Theorem}
\newtheorem{lem}[thm]{Lemma}
\newtheorem{rmk}{Remark}
\newtheorem*{defn}{Definition}
```

Two things to note here. The `[thm]` in the line defining Lemmas, says that Lemmas and Theorems use the same numbering scheme. The `\newtheorem*` when creating the Definition environment, means that definitions aren't labelled. Now I can create Theorem, Lemma, Remark or Definition environments with `\begin{thm}...\end{thm}`, etc. So, for example, when I type

```
\begin{defn} A number $p$ is prime if $p$ is not divisible by any
numbers other than 1 and itself.
\end{defn}
\begin{lem} 7 is prime. \end{lem}
\begin{thm} There are infinitely many prime numbers \end{thm}
\begin{rmk} There are also infinitely many non-prime numbers \end{rmk}
```

I get:

Definition. *A number p is prime if p is not divisible by any numbers other than 1 and itself.*

Lemma 1. *7 is prime.*

Theorem 2. *There are infinitely many prime numbers*

Remark 1. *There are also infinitely many non-prime numbers*

Note that we have Theorem 2, continuing the labelling from Lemma 1, but not Remark 3. Also the Definition is unnumbered.

8.6 Proof Environments

Another convenient environment is the proof environment, which is, unsurprisingly, traditionally used to contain proofs of things. Mostly, it just puts "proof" at the start and a box at the end. Produce a proof environment by putting things between `\begin{proof}...\end{proof}`. So for example,

```
\begin{lem} 4 is not prime \end{lem}
\begin{proof} 4 is divisible by 2, and therefore not prime.
\end{proof}
```

Gives us:

Lemma 3. *4 is not prime*

Proof. 4 is divisible by 2, and therefore not prime. □

You can also change what is said at the start of the proof by putting something else in brackets right after the begin statement. For example,

```
\begin{proof}[Proof of Lemma 1]
7 is not divisible by 2,3,4,5 or 6, nor any number
more than 7. Therefore, it is prime.
\end{proof}
```

Yields:

Proof of Lemma 1. 7 is not divisible by 2,3,4,5 or 6, nor any number more than 7. Therefore, it is prime. □

8.7 Enumerate Environments

These are used for making numbered lists. Between `\begin{enumerate}` and `\end{enumerate}` have text. Use `\item` commands to separate different items. For example,

```
\begin{enumerate}
\item This is an enumerate environment.
\item See the items.
\item Itemize is similar but without numbers.
\begin{itemize}
\item You can also nest them.
\end{itemize}
\item You can also add equations like $x+y=z$.
\end{enumerate}
```

produces:

1. This is an enumerate environment.
2. See the items.
3. Itemize is similar but without numbers.
    - You can also nest them.
4. You can also add equations like $x + y = z$.

9 Labels and References

One of the nice things about LaTeX's theorem and equation numbering is that it makes it easy to refer to a particular one. To do this you need to add a label command at the top of the theorem or equation environment, of the form `\label{<name>}`. You can later use the command `\ref{<name>}` to print the number associated with that theorem or equation. Using `\eqref{<name>}` puts it in parentheses. For example, if we have:

```
\begin{thm}\label{eulerTheorem}
\begin{equation}\label{eulerEquation}
e^{\pi i} + 1 = 0
\end{equation}
\end{thm}
We can use Theorem \ref{eulerTheorem} to evaluate
$e^{\pi i}$. In particular, subtracting 1 from both
sides of Equation \eqref{eulerEquation} tells us that it is -1.
```

it produces:

Theorem 4.

$$e^{\pi i} + 1 = 0 \tag{2}$$

We can use Theorem 4 to evaluate $e^{\pi i}$. In particular, subtracting 1 from both sides of Equation (2) tells us that it is -1.

10 Bibliography and Citations

If you want to have references you need a bibliography. The standard way to do this is via a bibliography environment. You create it as follows:

```
\begin{thebibliography}{<large number>}
<stuff>
\end{thebibliography}
```

The number of digits in the large number above tells LaTeXhow many digits to use in its bibliography numbers, so make sure it's big enough (I usually just use 99). Within you have bibitems. The syntax is `\bibitem{<citation key>}`. Each bibitem corresponds to one reference, producing a bracketed number (which you presumably follow with whatever description you want to write about it). The citation key is used to reference that item using a `\cite{<citation key>}` command, which works basically the same way as the `\ref` command. For example, this paper's bibliography section looks like this (the `\url` commands help create links, you need the hyperref package to get them):

```
\begin{thebibliography}{99}
\bibitem{aops} The Art of Problem Solving Wiki \LaTeX Guide
\url{http://www.artofproblemsolving.com/Wiki/index.php/LaTeX:About}
\bibitem{symbols} The Art of Problem Solving Wiki \LaTeX Symbols
\url{https://www.artofproblemsolving.com/Wiki/index.php/LaTeX:Symbols}
\bibitem{bib2} Ki-Joo Kim \emph{A BibTex Guide via Examples}
\url{http://www-hep.colorado.edu/~jcumalat/4610_fall_10/bibtex_guide.pdf}
\bibitem{this} \LaTeX source for this guide
\url{http://cseweb.ucsd.edu/~dakane/CSE101/latexGuide.tex}
\bibitem{bib1} Martin Osborne \emph{Using BibTeX: a short guide}
\url{http://www.economics.utoronto.ca/osborne/latex/BIBTEX.HTM}
\bibitem{miktex} MiKTeX Homepage \url{http://miktex.org/}
\bibitem{test} This is just a test, and not a real reference.
\bibitem{winedt} WinEdt Homepage \url{http://www.winedt.com/}
\end{thebibliography}
```

And the results are shown below:

References

[1] The Art of Problem Solving Wiki LaTeXGuide http://www.artofproblemsolving.com/Wiki/index.php/LaTeX:About

[2] The Art of Problem Solving Wiki LaTeXSymbols https://www.artofproblemsolving.com/Wiki/index.php/LaTeX:Symbols

[3] Ki-Joo Kim *A BibTex Guide via Examples* http://www-hep.colorado.edu/~jcumalat/4610_fall_10/bibtex_guide.pdf

[4] LaTeXsource for this guide http://cseweb.ucsd.edu/~dakane/CSE101/latexGuide.tex

[5] Martin Osborne *Using BibTeX: a short guide* http://www.economics.utoronto.ca/osborne/latex/BIBTEX.HTM

[6] MiKTeX Homepage http://miktex.org/

[7] This is just a test, and not a real reference.

[8] WinEdt Homepage http://www.winedt.com/

I can cite items above like so:

```
Note that \cite{test} is not a real paper. But you would know
that if you read \cite{this}.
```

which produces:
Note that [7] is not a real paper. But you would know that if you read [4].
It should be noted that there is another way to do bibliographies known as BibTeX. It is somewhat more involved, but allows you to produce your entries in a more standardized way. If you are interested in learning about it you may consult [5] or [3], or just Google for a guide yourself.

11 Macros

You can also define new commands. Typically, these are macros that allow you a shortcut for producing something common. To make them, put a `\newcommand` line near the top of your document. The general syntax is:

```
\newcommand{\<command name>}{<command output>}
```

Then, later in the document typing `\<command name>` will essentially be replaced with `<command output`. For example, this document has

```
\newcommand{\Z}{\mathbb{Z}}
```

at the top. Now when I type `$\Z$`, it produces the output $\mathbb{Z}$.
It is also possible to define commands with parameters, but that is beyond the scope of this tutorial.

12 Final Remarks

Hopefully this will be enough to get you started. If you have any questions feel free to ask me (though if it's anything particularly complicated, I might not be much better than Google). If you want an example of a .tex file, you can find the source for this guide at `http://cseweb.ucsd.edu/~dakane/CSE101/latexGuide.tex` ([4]).