A Hands-on Introduction to Creating Mathematical Documents with LaTeX

John Adamski

March 22, 2023

Abstract

Have you ever wondered how professional mathematical documents are created? How the equations and symbols can be positioned and formatted so beautifully? The answer is that these documents are created very differently from Word documents. They are created using a mark-up language (code) called LaTeX and then compiled to produce the final document. In this workshop, I will give a brief overview of LaTeX, explain different options for how to work with LaTeX files, and share some samples for attendees to play with and use to create their own documents. Everyone is highly encouraged to bring a laptop or tablet to this event.

Contents

1 Latex is a mark-up language . . . 1
1.1 Spaces, newlines, and paragraphs . . . 2
1.2 Comments . . . 3

2 Sections and environments . . . 3
2.1 Math environment . . . 4
2.2 Math fonts . . . 6
2.3 Parentheses . . . 6

1 Latex is a mark-up language

A lot of document editing programs like Microsoft Word and Google Docs are WYSIWYG – What You See Is What You Get. In those programs, you type and create the final document directly. If you want to skip a line, you just skip a line. LaTeX does not work this way. LaTeX is a mark-up language (like HTML) that you use to create a plain text file that is later compiled to produce your final document. In other words, you do not create the final document; you create a set of instructions for how you would like the final document to be created. And the beauty of it is that you do not need to (and sometimes cannot) worry about the tiny details of formatting – this is largely handled automatically by the compiler.

By default, everything we type into the plain text `.tex` file will be printed on the final document except for the nine characters below.

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

Those are special charcters that let the compliler know that what follows is not content to be printed, but instructions for the compiler. More on that later.

1.1 Spaces, newlines, and paragraphs

Notice that no matter how many spaces I put in my `.tex` file, the printed document includes only one. You do not control the space between characters, the compiler handles this, and it does it well (extra space after periods, hyphenated line-breaks, etc.). Notice that newlines are not printed how you type them. If you want to start a newline, you need to put `//` at the end the line, and then start a new line below.
Just
like
this. If you want to start a new paragraph, just put a blank line in your code.

Like this. Skipping more lines in your code…

will not have any effect. If you want to insert more space between sections of text, use the command `\vspace{1in}`, where the argumanet 1in can be replaced with any number of any of the following units.

cm	Centimeters
em	The width of the letter M in the current font
ex	The height of the letter x in the current font
in	Inches
pc	Picas (1pc = 12pt)
pt	Points (1in = 72pt)
mm	Millimeters

See that? ↑ That's a table. The `{ll}` argument lets the compiler know that my table has two columns and that each column should be left-justified. Within the table, the

`&` sign is used to separate columns, and the newline command `\\` is used to start a new row. You can learn more about tables by googling “latex tables”`latex tables`[1]. Notice that since I didn’t leave a blank line after the table, a new paragraph was not started.

That requires a blank line.

1.2 Comments

Whenever we write code, it is good style to insert comments. Anything in a line of code following a `%` sign is entirely ignored by the compiler. We insert comments in our code for our own benefit. For example, it can be to remind us what a block of text is. Commenting out some of your code can also help you when debugging (there will be bugs). Just insert `%` signs at the beginning of any line you wish to have the compile ignore. This way you do not have to delete it. It will still be there when you want to try including it again.

2 Sections and environments

I think you can see by example that we use the `\section{title}` and `\subsection{title}` commands to spilt our document into sections and subsections. Numbering of sections is automatic. If you don’t want a section or subsection to have a number, then use the `\section*{title}` and `\subsection*{title}` commands. The `*` will suppress the number.

One of the most useful features of LaTeX is the ability to `\begin` and `end` different environments. Right now, we are not in any environment and my text is rendered in the usual way.

But I can have my text centered by entering a new envirnment for that.

There are even environments for theorems, definitions, proofs, etc.

Theorem 1 (My silly theorem)**.** *This is a silly theorem.*

Proof. Don’t you see? It’s silly! □

Definition 1. This is the definition environment.

You can even make your environments like these. To see how this is done, scroll to the top and look at the block of code labelled `%ENVIRONMENTS`.

[1] Duh. But really, the internet is very helpful. You will find answers to pretty much any LaTeX question you have on urlhttps://tex.stackexchange.com/.

2.1 Math environment

The one you've been waiting for. To enter/ext in-line math mode, we use the $ sign. Here is an example: $\sin^2\theta + \cos^2\theta = 1$. This is good for smaller things, short equations or referencing variables x, y, z. For bigger equations, you will want to use the equation environment.

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi}{6} \tag{1}$$

Within the math environment, you have access to many many many many symbols. Here is a reference for some of the symbols you can use. Here is a web app called Detexify that lets you draw the symbol you want and tell you what the LaTeXcommand for tha symbol is.

By default, exponents are assumed to be one digit/character/symbol. Things like 3^2 and e^x works just as expected. But if your exponent is more than a single character, you will need to use brackets {} around your exponent so that LaTeXknows to put all of that stuff in the exponent position. Suppose you want to raise 2 to the 10th power. You cannot write 2^10 as you would with a calclator[2]. You must write 2^{10}.

In general, attaching an underscore _ to a symbol with cause the next symbol to be a subscript, ad attaching a carot ^ will cause the next symbol to be a superscript.

$$T(x) = \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

It is even possible for subscripts and superscripts to have subscripts/superscirpts.

$$x_{\alpha_0}^{{\beta_0}^{2^{10}}}$$

But you should probably reconsider your choice of notation if you find yourself typing such things.

Here are some additional equations that demonstrate some other LaTeXcommands.

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

[2] Maybe here is a good time to mention that LaTeXis not checking your math, or doing anything of the sort. It will not let you know when you leave out parentheses or anything like that. It is just printing symbols.

Notice that the & symbol is used to separate columns, just as it was used within the \tabular{} environment. It can also be used to create matrices.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 & 9 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

$$f(x) = \begin{cases} 3x+4 & \mbox{if } x < 1 \\ \ln x & \mbox{if } 1 \leq x < 4 \\ \sin x & \mbox{if } x \geq 4 \end{cases}$$

The \mbox{} command can be used to print regular text within a math environment. When you try to write regular text in a math environment *without* using \mbox{}, it looks very weird.

$$This is some regular text printed inside math mode$$

$$12 \equiv 5 \pmod{7}$$

$$\emptyset \subseteq A \subset A \cup B$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

$$|X| < \infty \Rightarrow |\mathcal{P}(X)| = 2^{|x|}$$

$$\lim_{x \to a} f(x) = f(a)$$

$$\frac{d}{dx} \int f(x)\ dx = f(x)$$

$$\int_1^x \frac{1}{x}\ dx = \ln x$$

2.2 Math fonts

You can changee the font of letters using the following commands.

1. `\mathbb{Z}` → $\mathbb{Z}$
   This is the font for all the familiar sets of numbers: $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$

2. `mathcal\{P}` → $\mathbb{P}$
   This is a caligraphy font that makes letters just a little fancy: $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \ldots$

2.3 Parentheses

Sometime we want parentheses to be big. For example, when the expression they contain is big. If we use `\left(` and `\right)`, then the parentheses will be scaled to match the size of what appears between the two commands. See the difference below.

$$(\frac{\pi}{3}) \text{ vs. } \left(\frac{\pi}{3}\right)$$