LaTeXtutorial

Alejandro Rivera

October 2020

Introduction

These notes are far from being a complete introduction to Latex. A good way to get started with writing Latex is to have a concrete writing project to work on and these will come soon. The purpose of these notes is just to show you the first steps of using the language. I encourage you to look online if you have any doubts. There is a vibrant online community. You can look for documentation and/or help here or elsewhere:

- `https://mirrors.rit.edu/CTAN/macros/latex/required/amscls/doc/amsthdoc.pdf`
- `https://tex.stackexchange.com/`
- `https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols`

1 The origin and basic principle of Latex

Latex is a programming language derived from Tex, which was created by Donald Knuth in 1978. Tex was expanded to Latex by Leslie Lamport in 1984. At the time, text editing programs were quite rudimentary and Tex brought many innovations to the community.
You may be familiar with text editing programs such as Libre Office for example. Contrary to typical text editors, in Latex, you do not work directly on something that looks like the output. Instead, you work on a **code**: a text file whose name ends with .tex; and you **compile** this code to obtain a **pdf output**. The text file may be edited using generic notepad editors and the compilation can be done separately, but there are programs that do everything at once, such as *emacs* or *Texmaker*. You can install these or others on your computer depending on your operating system. In this tutorial, we will be working with *Overleaf*, which is an online latex editor with document sharing capabilities.

2 Getting started with Overleaf

Here is how to set up a project on Overleaf, where you can write in Latex and export pdf documents.

- Go to `www.overleaf.com`, create an account and login.
- On your dashboard, click on "New Project", select "Blank project" and name your new project.
- As you can see the screen is divide in two zones : on the left is the code, on the right is the output. You can edit the name of the author by writing in the \author{...} area in the code.
- When you are done editing, save and compile using Ctrl+S. The result of your editing should appear in the document on the right.
- At any time you may download the pdf output by clicking on the Download button on the top right. You may download the code or share it with your friends from the Dashboard.

3 Writing basics

Here are the most basic things you can do when writing math in Latex:

- Try writing some text below \maketitle. Notice that Latex does not notice when you skip lines. To skip a line in the output, use \\.
- Add \usepackage{amsmath} in line 3 of the code and \usepackage{amsfonts} below it. These commands call libraries that allow you to use math symbols in your document.
- To write your first equation, go back to where you were writing text and, in a new line, insert \begin{equation}. As prompted, add the corresponding "end equation" command below.
- Inside the box, write f(x)=1+1. The result should look like this:

$$f(x) = 1 + 1. \tag{1}$$

- As you can see, inside the equation environment, things work very differently from the text environment. This is a key distinction in Latex.

In the rest of the present document, instead of telling you what to write, I suggest you look for ways to implement various commands. The first one is how to write things in bullet points the way I have done so far. The name of the command is **itemize**. See if you can find how it works online by trial and error.

4 Some useful math symbols

The most important thing to understand about the math environment is the role of curly brackets { and }. These play the role of parentheses in the code: they cause their contents to behave as a group with respect to other operations.

- Look for a way to write indices A_1, A_2, A_3 and exponents $1 + X^2 + 3X^4$.
- Once you have figured this out, you may want to try something more ambitious like X^{Y^2+1}. To do this, put the exponent between $\{\dots\}$. What happens if you forget to do this?
- In math, we love greek letters. These are spelled out as \alpha for example, which looks like α. There are also τ, Υ, Ξ and many more.
- In our exercise sheets, we use fancy letters like $\mathcal{F}$ or $\mathbb{P}$. These are special fonts. For instance $\mathcal{F}$ is produced using the **mathcal** font. There is also the **mathfrak** font, the **mathbb** font and several others. See if you can reproduce these symbols by looking up these keywords.
- Say you want to do some set theory. Find a way to write $A \cap B \neq \emptyset$.
- Maybe you'd rather do some analysis. What about $\forall \varepsilon > 0$, $\exists \delta > 0$... Can you find how to write quantifiers and comparisons like $<>, \leq, \geq$?
- Or maybe you prefer algebra and you want to write $\frac{X^3+1}{X+\frac{X^5-2}{4}}$. Fractions are written using the \frac{...}{...} command. can you figure out how it works?
- There are many other math symbols you can write, like matrices, limits, sums, integrals, sin, cos, exp, ln, commutative diagrams... I'll let you explore on your own according to your needs.

Try to write $e^{i\pi} = 1$. How about $e^{i\pi} = -1$? You'll notice that everything compiles fine although the math is wrong. This is because **Latex doesn't care about math**. It is just a text editor.

5 Math environments

As you can see, I have been using inline math environments like $x^2 + 1 = 2$ inside my sentences. This is not possible with the first command I gave you. To open a math environment inside a text paragraph, just put it between dollar signs (one on either side).

You might also want to write very long equations. For this you can use the **multline** environment. Or if you want to do a very long computation, **align**

can be very helpful. Try reproducing the computation below:

$$\begin{aligned} 1+1+1+1+1+1+1+1+2 &= 2+1+1+1+1+1+1+2 && (2)\\ &= 2+2+2+2+2 && (3)\\ &= 4\times 2 && (4)\\ &= 8\,. && (5) \end{aligned}$$

You may want to refer to previous equations later on in the text. For instance, maybe Equation (1) could come in handy right now. You could try writing the reference by hand, but this could be tricky. What if you later switch the orderings of your equations? In order to ensure that the code keeps track of your equations even if you move them around the text, you can label them. Look up **labeling equations** in latex. This is a simple way to name your equations so you can refer to them at any time while writing, wihout worrying too much about their position in your document.

6 Theorems and proofs

In the course notes, there are Theorems, Lemmas and Propositions. These have statements and proofs. Here is how to use them in your Latex project:

- Add the following command below \usepackage{amsmath}: \usepackage{amsthm}.
- Now, add some space between the \usepackage{...} commands and the \title{...} command, and insert the two following commands, each on a new line: \newtheorem{theorem}{Theorem}, \newtheorem{exo}{Exercise}.
- Now, back to the bottom of the Latex document (just above \end{document}). You can define a theorem as follows. Begin with begin{theorem}[There was], skip a few lines and write end{theorem}. Now, between these two delimiters, you can write the statement of your theorem. The result should look something like this.

  Theorem 1 (There was)**.** *.... and old lady who lived in a shoe.*

- You can label theorems the same way you label equations.
- In order to write a proof of your theorem you can use the begin{proof} end{proof} environment:

  Proof. **This is just to say**
  I have eaten
  the plums
  that were in
  the icebox
  and which

you were probably
saving
for breakfast
Forgive me
they were delicious
so sweet
and so cold

–William Carlos Williams □

- If you would like to use the proof environment to write answers to exercises, you can just write \begin{proof}[Answer] instead.
- As an example, try reproducing the following :

  Theorem 2 (My topology theorem)**.** *Let (X, τ) be a topological space. Let $f : X \to \mathbb{R}$. Then, f is continuous if and only if $\forall t \in \mathbb{R}$, $f^{-1}((t, +\infty))$ and $f^{-1}((-\infty, t))$ are open.*

  By Theorem 2 the function $f : x \mapsto x^2 + 1$ is continuous.

  Proof of Theorem 2. Let $s, t \in \mathbb{R}$ be such that $s < t$. Then, $f^{-1}((s, t)) = f^{-1}((s, +\infty)) \cap f^{-1}((-\infty, t))$ which is open. Therefore, for any open interval I, $f^{-1}(I)$ is open. Let U be an open subset of $\mathbb{R}$. Then, for each $x \in U$, we may find an open interval I_x such that $x \in I_x \subset U$. In particular, $U = \cup_{x \in U} I_x$ so $f^{-1}(U) = \cup_{x \in U} f^{-1}(I_x)$ which is open. Hence, f is continuous. □

7 Macros

This is where Latex really starts getting interesting. As you may have noticed by now, the code is divided into a first part, which is called the preamble, and a second part, containing the body of the text, delimited by \begin{document} and \end{document}. The preamble behaves much more like code while the document corresponds more or less to what will be displayed on the .pdf file. The preamble is very powerful: you can call on packages containing new commands using \usepackage{...} or even write your own. When you write your own command, this is called a macro. Here is how it works in the simplest case:

- Near the end of the preamble, add \newcommand{\mathcal{D}(U)}{\myset}. When you add this command to your preamble, you can write \{myset} in the text and the compiler will interpret this as \mathcal{D}(U) and output $\mathcal{D}(U)$. A common macro is the following:

  \newcommand{\R}{\mathbb{R}}

  which is really handy when doing analysis on $\mathbb{R}$.

- There are more sophisticated macros that allow for variables and parameters. They are presented in the standard documentation in case you need them.
- Sometimes, the macro you are looking for has already been written by someone. There are many posts online discussing ways to implement this or that macro.