LaTeX for the Constantly Busy

Nathan Chapman

December 25, 2016

I dedicate this work to the people who strive for truth about our reality, otherwise known as scientists.

"Math: a lifetime of free entertainment" - Dr. Branko Ćurgus

Preface

Unlike other word processors (Word, Google Docs, Open Office, etc.), LaTeX is specifically meant to be a technical, scientific writing software. Also unlike other word processing software, LaTeX is not a "What you see is what you get" piece of machinery, but rather a It provides a clean, completely customizable way to present your work. In this guide, we will see the concepts and code behind beautifully curated documents that should be an art all on their own.

The biggest problem with learning LaTeX is the enormous learning curve there is. When you're first starting, unless you're a LaTeX prodigy, you are going to struggle and get loads of error every time you compile, and it **will** take forever. The only to have this not happen, is to keep doing it and eventually you'll get the hang of it. I believe in you. Once past this point, you will be able to almost effortlessly create gorgeous documents that are full customized to your liking.

Note: This was written in Texmaker, which is what I like and use, version 4.5, and so the non-code parts might be exclusive to this program. There are other LaTeX programs that do the very same thing, after all, it's really just a front-end compiler. A very popular, and online LaTeX document editor/creator is *ShareLaTeX*. This is also used for groups who, as the name would suggest, share a document and edit together. Other editor/creators are easily available on the internet.

Note 2: When searching online for documentation on LaTeX , I highly suggest both the LaTeX stackexchange website, the *ShareLaTeX* documentation, and the LaTeX Wikibooks site. All of these have well done documentation, and are usually very helpful.

Contents

1 Style

Before we begin, I would like to point out the importance of the style of your code. If you have coded before, that's great and you probably already know how you like to style it. If you haven't, that's fine as well. I will suggest my stylistic layout for LaTeX code, but that is merely a suggestion because you can do whatever you want as long as it works for you (Though if you're working with other people on a document, it would be better for everybody if the code is written in a way that everyone can at least tolerate and read).

The way I like to write my code, not just for LaTeX but other languages as well, follows a very "logic tree" format. By this I mean, every subset in a set will start a single, four-space tab to the right of the indentation of the set. For example

```
\part{Important Functions}

‘‘tab’’ \chapter{Plot}

‘‘tab’’ ‘‘tab’’ Here we will look at more of the customization options for Plot.

‘‘tab’’ \chapter{Manipulate}

\part{Graphics}
```

where each "tab" refers to a single, four-space tab. This way, it is quick and easy to read and find code.

2 The Heading

The heading is where all your "imports" and new commands and environments go. The easiest and quickest way to do a heading that will probably suit your needs is with the wizard *Quick Start*. If you do not have this option, then here is probably a safe bet as to what you should have (minus the numbers to the left).

```
(1) \documentclass[11pt,letterpaper,twoside]{article}

(2) \usepackage[utf8]{inputenc}
(3) \usepackage{amsmath}
(4) \usepackage{amsfonts}
(5) \usepackage{amssymb}
(6) \usepackage{graphicx}
(7) \usepackage[in]{fullpage}
```

From top to bottom

1. This goes at the very tippy-top of your document, before anything else. This tells LaTeX you want your document to be of the *article* class, have 11 pt font, formatted for letter sized paper (the standard in the U.S.), and formatted to be printed double-sided.

2. This has to do with the input from the user, and deals with fonts. If you're strictly typing in English, you don't need to deal with this. If you are going to be using characters that are not included on the standard English keyboard, you should also include "\ usepackage[T1]{fontenc}".

3. This is a very important package to include if you're going to be using really, any math in your document. This package comes from AMS, the American Mathematical Society. It allow you to do things like use math mode and such.

4. This package allows the user to use special fonts like "mathcal" $\mathcal{F}$, or "mathbb" (The "bb" stands for blackboard) $\mathbb{R}$, and more.

5. With this package, you can even use symbols commonly used in math texts. For example, "odot" $\odot$, or the entire lowercase and capital greek alphabet.

6. The "graphicx" package allows you to input pictures into your document, as well as format them and their surrounding to suit your needs.

7. This package doesn't come standard with the wizard, but I feel it is very important to have; otherwise, you have giant margins and waste a lot of paper. Here I used the "[in]" setting to place one inch margins on each side of the page, where I could have used "[cm]" to place, you guessed it, one centimeter margins on each side of the page.

3 The Body

Simply put, the body is where you write what you want after you know how to do it. This part of the document comes in between the "begin/end{document}" commands. Here you make your title if you want, make a table of contents if you want, and write what you want to write. Basically, if it's not in the preamble, it's in the body. That being said, there are at least two things worth mentioning, sectioning, and tabulating.

3.1 Sectioning

Sectioning in LaTeX is pretty easy, and makes sense since it follows the normal hierarchy of sections. Starting from the outer most type of section to the innermost we have: Part, Chapter, Section, Subsection, Subsubsection, Paragraph, Subparagraph.

Note: "Part" and "chapter" sections are only allowed in the "book" or "report" document class.

4 Math Mode

There are several ways to use math mode. Depending in what context you are using math, you can choose display math, inline math, or equation/align environments.

4.1 Display

Display math mode is perfect for clearly showing an equation or some significant mathematical statement. You can invoke this by enclosing the statement *expr* in

```
\[ expr \]
```

which will invoke the ability to use math symbols, center the text and italicize it.

For example, the general form of the total derivative for a function $y = f$ whose variables $x_1, x_2, \ldots, x_N$ are also T-times-composed functions of t is

$$\frac{\partial y}{\partial t} = \frac{\partial f}{\partial t} + \sum_{k=1}^{N} \left[f'(x_{k,1..T}) \prod_{i=1}^{T} x'_{k,i..T}(x) \right]$$

or the general Fourier cosine and sine series of a function f in x is

$$f(x) \sim a_0 + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi}{L}x\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi}{L}x\right)$$

4.2 Inline

If you want a mathematical obj *obj* in math format, but also inline with the rest of the text, you would use

```
$ obj $
```

inline with the rest of the text. The code for this environment is the same as if it were in display math mode. You can also force LaTeX to show the math in display style, while still being inline with the text with

```
$ \displaystyle obj $
```

For example, the general form of the total derivative for a function f whose variables $x_1, x_2, \ldots, x_n$ are also T-times-composed functions of t is $\frac{\partial y}{\partial t} = \frac{\partial f}{\partial t} + \sum_{k=1}^{n} \left[f'(x_{k,1..T}) \prod_{i=1}^{T} x'_{k,i..T}(x) \right]$, and the general Fourier cosine and sine series of a function f in x is $f(x) \sim a_0 + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi}{L}x\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi}{L}x\right)$.

4.3 The Equation Environment

The equation environment does the same thing as the the centered displaymath mode, but now it numbers the equation off to the right side of the page. This is great for when you are using and referencing a lot of equations.

For example, to number a mathematical statement *obj* and display it, we use

```
\begin{equation}
obj
\end{equation}
```

4.4 The align* environment

In the case where you have multiple equations that you need to neatly present, it is most often the case where you should align them in some way.

To align several equations, in display math mode, or whatever you want really, we use

```
\begin{align*}
expr1 &= expr2 \\
expr3 &= expr4
\end{align*}
```

Here the “&” is used to tell LaTeX where to align the expressions, and the “\\” tells LaTeX to start a new line.

Note: The last equation does not need a “new line” command, thus it can be omitted. If it *is* inserted, it will compile and output as if there wasn’t one.

Note: The “*” in the beginning of the align environment tells LaTeX to not number the equations, or whatever you have in there. To have the equations numbered, simply omit the star.

4.5 Superscripts and Subscripts

If we want x^α, then we use, while in math mode,

```
x^\alpha
```

or if we want $x^{\alpha+\beta}$, we use

```
x^{\alpha + \beta}
```

The curly brackets around this superscript argument are very important, because when the curly brackets are omitted, only the first character is raised.

Similarly, the same is for subscripts as it is for superscripts. e.g. If we want x_α, we use

```
x_\alpha
```

and for the longer argument,

```
x_{\alpha + \beta}
```

4.6 Fractions

Eventually, more likely than not, soon, you will encounter a fraction that you want to write. In LaTeX , if you want the fraction $\frac{a}{b}$, then you use

```
\frac{a}{b}
```

or if you want to have fraction in line, but force LaTeX to show it in display style, you use

```
\dfrac{a}{b}
```

4.7 Brackets

Usually you have a fraction in your math that you need to display but you also have some sort of bracket around it. Whether it be parenthesis, curly brackets, square brackets, angle brackets, etc., all you need to do is prepend a backslash then the side next to the character.

For example if we have $\cos(\dfrac{n\pi}{L})$ and we want $\cos\left(\dfrac{n\pi}{L}\right)$, then we use

```
\cos \left( \dfrac{n \pi }{L} \right)
```

4.8 Spacing

It is built into LaTeX that, when in math mode, spaces between characters are not rendered in the output document. This comes up in the super/sub-script command when using more than one character. At first this may seem frustrating, and it is at that point, but after a while of using LaTeX , you will see this is actually very helpful and a blessing. Nevertheless, when spacing is needed, we can still take care of it.

There are a few different ways to space in math mode, and they all correspond to an obscure way of typography. The spaces we will see are based of the "mu" unit which is, according to *ShareLaTeX*,

> *math unit equal to 1/18 em, where em is taken from the math symbols family*

and the "em" unit is

> *roughly the width of an 'M' (uppercase) in the current font (it depends on the font used)*

Anyways, once used enough, you will know what you want to use in different situations.

- `\,`

  gives a small space, e.g. $(\,)$, (3 mu).

- `\:`

  gives a medium space, e.g. ( ), (4 mu).

- `\;`

  gives a large space, e.g. ( ), (5 mu).

- `\quad`

  gives a *space equal to the current font size*, e.g. (   ), (18 mu).

4.9 Special Characters and Operators

LaTeX has built in commands for math functions, so you don't need to do some irritating typesetting every time you try to use text in math mode. LaTeX is also very user friendly, with respect to naming commands, because just about every command is exactly what you would hope it would be, its name. This even applies to the case of the letter, or the size of the symbol, or the behavior of the symbol. For example,

- `\pi`

  gives “π” when in math mode, while

  `\Pi`

  gives “Π while in math mode.

  Note: This is not the same notation you use for indexed products.

- `\Sigma`

  gives “Σ” in math mode, but

  `\sum`

  gives “$\sum$” in math mode. The difference here being the “sigma” character is the capital Greek letter “Sigma”, while the “sum” command is telling LaTeX to typeset this mathematical statement using summation notation.

  Similarly for “alpha” and “propto”,

  `\alpha`

  produces “α”, while

  `\propto`

  results in “$\propto$”.

- `\cup`

  results in a character representing a cup $\cup$, while

  ```
  \bigcup
  ```

  signifies the mathematical operation $\bigcup$.

The moral of this story is

to make LaTeX produce the symbol you want, literally type a backslash, then the name of the thing you want

whether it be an operation, or character, and LaTeX will probably already have it in its library and do exactly what you want.

4.10 Limits

Sometimes your operators have their limits off the side, but that doesn't look as good as if they were typeset to their full area.

To make your stuff nice, like having your limits/indicies, we use

```
\operator\limits_{lb}^{ub}
```

where "operator" is the operator, "lb" is the subscript on the operator, and "ub" is the superscript on the operator.

For example, in inline math mode, the summing $x_1, x_2, \ldots, x_N$ using

```
\sum_i^N x_i
```

produces $\sum_i^N x_i$, while

```
\sum\limits_i^N
```

yields $\sum\limits_i^N$.

Note: When using an operator in display math mode, the limit command is usually omitted and done automatically.

5 Useful Tools

Though all of LaTeX is useful, I think there are a few especially important features and functions that should have their own light shed on them.

5.1 enumerate

The enumerate environment is a way to nicely number things in a list, as in the "Heading" description. It is a great environment for students because it makes formatting homework assignments and such, as simple as listening to a soft-voiced, Canadian man teach electronics.

The enumerate environment works like this

```
\begin{enumerate}

\item %1

\begin{enumerate}

\item %a

\begin{enumerate}

\item %i

\end{enumerate}

\end{enumerate}

\item %2

etc.

\end{enumerate}
```

This same exact structure also applies to the "itemize" environment. This environment does the same thing as enumerate, but instead of numbers, there's bullets.

Note: The document won't compile if there isn't at least one *item* in the enumerate environment.

5.2 graphicx

The "graphicx" package that needs to be imported in the preamble gives the user the ability to insert pictures into the document.

To insert a picture into the document, we use

```
\includegraphics[scale=●]{●}
```

You can replace the "scale" part of the command with other things, like "width". A very common and useful substitution of this is

```
width=x\textwidth
```

where "x" is some number; this says the width of the picture is x times the width of the text. In the curly brackets, you put the file address of the picture you would like to insert. If the picture file is in the same folder as the LaTeX document, the name of file **with extension** can replace the bullet in curly brackets. It is also best if the **file name does not have spaces in it**, as LaTeX was programmed in time when computers weren't good at handling spaces in file names.

5.3 Tabular

There will come a time when it will be just about necessary to format a set of objects into a table format, for instance a matrix, but don't fret, you have your handy-dandy guide with you to help you learn how to do it.

The best way to make a centered, table of 3 columns, where the content in each cell of the table is centered, in LaTeX is

```
\begin{center}
\begin{tabular}{ (1) |c|c|c| }
(2) \hline
(3) cell1 (4)& cell2 & cell3 (5)\\
 cell4 & cell5 & cell6 \\
 cell7 & cell8 & cell9 \\
 \hline
\end{tabular}
\end{center}
```

which produces

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Here

1. the " | " with the "c"s tell LaTeX to insert a vertical line all the way down the column.

2. the "hline" command tell LaTeX to insert a horizontal line across the entire table at that position.

3. each "cellX" is the content of that cell.

4. each "&" tells LaTeX where the right border of the column is, and also where to align the columns.

5. each “\\” tells LaTeX that is the end of the line, and to start a new one.

Now of course, you can indicate however many columns you want, with whatever alignment you want (“l” for left-aligned, “c” for center-aligned, and “r” for right-aligned), and whether or not to add horizontal line separators. For more information, I gladly point you toward the wikibook “LaTeX/Tables” page. They have a lot more information there, along with examples.

Conclusion

Whatever editor you're using, and whatever you're constructing, I urge you, don't give up. It will come out great, if you just give yourself more time, consult some online literature (even though its kinda hard to find explicit answers for some questions), and ask someone around your department who knows LaTeX (I can almost guarantee there is at least one, student or staff). Who knows, maybe eventually you will like the program so much and become proficient enough at it to write your own guide!

Nathan Chapman
Physics Sophomore
Math Senior
Western Washington University
December 25, 2016