Graduate Student LaTeX Tutorial

Michael A. Herrera

February 2016

1 Introduction

This tutorial was made for beginning users who don't have any experience in using LaTeX. This is not meant to be a comprehensive tutorial, but hopefully at the end, you will have enough information to continue learning on your own. There are many different ways to accomplish similar tasks in LaTeX and everything I know about LaTeX is self taught, so the examples that I provide may not be optimal. I have also made available the TeX file used to create this tutorial on my github account, https://github.com/mherr77m/LaTeX.

2 Installation of LaTeX

There are two parts to a LaTeX installation, the TeX distribution and the LaTeX front end. LaTeX is a programming language, not a stand alone program. The TeX distribution contains the source code for the language and a LaTeX front end is where you will write and compile your code. There are TeX distributions for all major operating systems and can be found at the following web address:

https://latex-project.org/ftp.html

Once you have the distribution installed, you will need a front end where you can write your paper. For this tutorial, I am using TeXShop which can be downloaded from here:

http://pages.uoregon.edu/koch/texshop/obtaining.html

This editor is only available for Mac and is the same front end that is installed on the department computers. If you would like a different editor, or if you are using another operating system, there are many other editors available, such as Texmaker:

http://www.xm1math.net/texmaker/

The other option is to use an online LaTeX editor. These sites allow you to use LaTeX without having to install it on your machine. One example of these sites is Share LaTeX:

https://www.sharelatex.com

3 Staring a New Document

When you open TeXShop, you are provided with a blank window in which to start writing your paper, Fig 1. LaTeX is a compiled language, meaning that when you want to create a PDF of your paper, you will need to compile your code. In TeXShop, to compile your code, you use the "Typeset" button in Fig 1. There are different compiling options available in the pull down menu next to "Typeset", but for now, we will leave it set to the default "LaTeX" option. If you aren't using any references in your text, you just need to compile the code twice, meaning that you use the "Typeset" button twice. This will link any equation and figure references you have in your text.

3.1 Preamble

The first thing that every LaTeX document needs is a *preamble*, which contains information that affects the entire document. The general structure of a LaTeX documents is,

```
\documentclass{...}
...
\begin{document}
...
\end{document}
```

Everything between the `\documentclass{...}` and the `\begin{document}` is called the *preamble*. This is where you will define your document type and load packages. There are many different document types, such as article, book, and letter, but for most of us we will be using article class. For this tutorial, your first command in your program should be

```
\documentclass[12pt]{article}
```

Each command in LaTeX is called by using a backslash `\`. Here, we have called the "documentclass" command and passed it two arguments. The argument in the curly brackets `{ }` is the class type and any arguments in the square brackets `[ ]` are the class options. There are multiple class options such as main font size, paper size, number of columns, etc. An additional example of the document class command is seen below.

```
\documentclass[10pt,a4paper,twocolumn]{article}
```

Another part of the preamble is the loading of packages. There are many packages included in the TeX distribution, but they are only loaded if you need them, very similar to importing in Python.

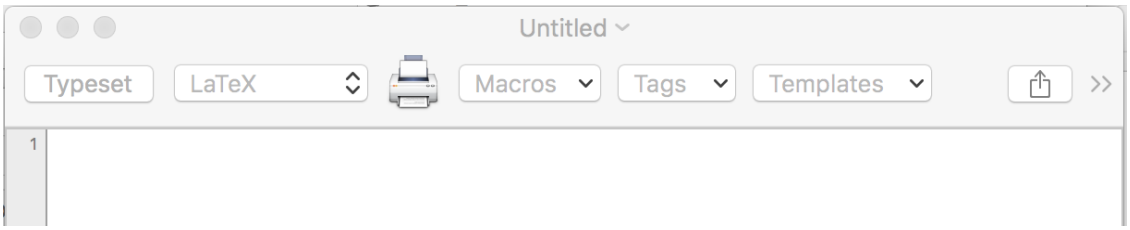

Figure 1: Example of the TeXShop environment.

One of the most used packages is graphicx, which allows you to easily put figures into your paper. If you want to change the margins of your paper, you can load the geometry package and set the margins. You can load these packages by,

```
\usepackage{graphicx}
\usepackage[margin=1in]{geometry}
```

where `\usepackage{...}` is the command, "geometry" is the argument passed to the command, and margin=1in is an argument specific to the geometry package.

3.2 Title Page

Now that we have defined our document and loaded a package, we want to add a title page to our paper. The first part of making a title is done in the preamble. For the title page in this tutorial, the following three commands were used in the preamble.

```
\title{Graduate Student \LaTeX \ Tutorial}
\date{January 2015}
\author{Michael A. Herrera}
```

Once you have defined the title, date, and author, you can call the `\maketitle` command within the body of the code,

```
\begin{document}
\maketitle
\newpage
\end{document}
```

The `\newpage` command will skip the rest of the page and start a new page. Everything that we want to add from here on will be added to the body of the code, between `\begin{document}` and `\end{document}`.

4 Body of Text

Once you have a title page, you can start the main body of your text. One of the best ways to organize a paper is by sections as is seen in this tutorial. The command to start a new section is `\section{Section Title}`, where you replace "Section Title" with the name of your section. After you define a section, you can write the body of your paper just as you would in any other word processor. If you want a section title without a number, you can add an asterisk to the command, `\section*{Section Title}`.

4.1 Sub-Section

You also have the option of using sub-sections (`\subsection{Sub-Section}`),

4.1.1 Sub-Sub-Section

and sub-sub-sections (`\subsubsection{Sub-Sub-Section}`).

4.2 Text Modifiers

There are many different ways you can modify the text of your paper, such as **Bold**, *italics*, or font size. I'm not going to go through every command you can use to alter text, but here is a list of some common commands:

Command	Result
`\textit{Italics}`	*Italics*
`\textbf{Bold}`	**Bold**
`\texttt{Teletype Font}`	`Teletype Font`
`\textsc{Small Capitals}`	Small Capitals
`\uppercase{Uppercase}`	UPPERCASE
`\underline{Underline}`	Underline
`\tiny Tiny`	Tiny
`\small Small`	Small
`\normalsize Normal Size`	Normal Size
`\Large Large`	Large
`\Huge Huge`	Huge

One thing to note about the font size commands is that you don't pass them any arguments. Instead, you call the command and then all of the text afterwards will be at the new font size. Once you want to return to the default font size, you can use the command `\normalsize`.

The last thing I wanted to show in this section is creating lists, such as bullet points and numbered lists. A simple bullet point list can be created using the following code,

```
\begin{itemize}
\item First bullet point
\item Second bullet point
\end{itemize}
```

which leads to,

- First bullet point
- Second bullet point

If you want a numbered list you can use "enumerate" instead of "itemize",

```
\begin{enumerate}
\item First bullet point
\item Second bullet point
\end{enumerate}
```

which leads to,

1. First bullet point
2. Second bullet point

Finally, you can nest and combine these two types of lists,

```
\begin{enumerate}
\item First bullet point
\item Second bullet point
\begin{enumerate}
\item Sub bullet point
\begin{itemize}
\item Sub sub bullet point
\item Sub sub bullet point
\end{itemize}
\item Sub bullet point
\end{enumerate}
\end{enumerate}
```

which leads to,

1. First bullet point
2. Second bullet point
   (a) Sub bullet point
      - Sub sub bullet point
      - Sub sub bullet point

   (b) Sub bullet point

5 Equations

Writing equations in LaTeX is very different than other word processors. Instead of using an equation editor to create an equation and then insert it into the paper, in LaTeX you write your equations using LaTeX commands. To load all of the needed symbols and expressions, you can load the mathtools package in the preamble of the TeX file, `\usepackage{mathtools}`.

5.1 In-Text Equations

Sometimes equations are simple enough that you want to place them in the text, $p = \rho RT$, or you can use this same technique to explain terms in an equation, ρ is density and p is pressure. To add an equation or use any math text within your text, you just need to surround the math expression in `$...$`. For example, the ideal gas law used earlier in this paragraph was created using `$p=\rho RT$`.

5.2 Separate Equations

The other way to write an equation is to place it on it's own line. This is usually easier to read and allows for the equation to be referenced other places in the paper. Using the ideal gas law from the previous example, we can rewrite the equation as,

$$p = \rho RT \tag{1}$$

You can see that the equation is now centered on it's own line and is numbered so that you can reference it in the paper. The code used for the above equation was,

```
\begin{equation}
p=\rho RT
\label{eq:one}
\end{equation}
```

The entire equation is wrapped in `\begin{...}` and `\end{...}` commands, with the actual equation in the middle. The other part of this equation is the label, which allows you to reference the equation later in the paper, Eq 1, `\ref{eq:one}`.

5.3 Symbols

There are a wide array of mathematical symbols and expressions available in LaTeX allowing you to write just about any equation you want. In this section, I've included some of the most commonly used expressions. To use these examples, you just need to place the expressions between the begin and end commands, like in Eq 1.

The most basic math symbols that are available on you keyboard can be used normally in an equation, such as,

$$+ - = ! / ( ) [ ] < > | ' :$$

There are two ways to use parenthesis and square brackets. If you you them normally, `( )` or `[ ]`, you can get an equation that looks like this,

$$\theta = T(\frac{p_o}{p})^\kappa \tag{2}$$

You can see that parenthesis are too small so to fix this issue, we can add the `\left` and `\right` commands before the parenthesis, `\left ( ... \right )`, which results in the equation,

$$\theta = T\left(\frac{p_o}{p}\right)^\kappa. \tag{3}$$

Greek letters are also available and easy to use, `\delta` becomes δ and `\Delta` becomes Δ.

`\alpha`	α	`\kappa`	κ	`\psi`	ψ	`\Delta`	Δ	`\Theta`	Θ
`\beta`	β	`\lambda`	λ	`\rho`	ρ	`\Gamma`	Γ	`\Upsilon`	Υ
`\chi`	χ	`\mu`	μ	`\sigma`	σ	`\Lambda`	Λ	`\Xi`	Ξ
`\delta`	δ	`\nu`	ν	`\tau`	τ	`\Omega`	Ω		
`\epsilon`	ϵ	`o`	o	`\theta`	θ	`\Phi`	Φ		
`\eta`	η	`\omega`	ω	`\upsilon`	υ	`\Pi`	Π		
`\gamma`	γ	`\phi`	ϕ	`\xi`	ξ	`\Psi`	Ψ		
`\iota`	ι	`\pi`	π	`\zeta`	ζ	`\Sigma`	Σ		

Fractions can be written using the `\frac{numerator}{denominator}` command.

```
\begin{equation}
 q = \frac{r_v}{1+r_v}
\end{equation}
```

Which produces,

$$q = \frac{r_v}{1+r_v} \tag{4}$$

To add a subscript you can use an underscore followed by the subscript, `_`. If your subscript is more than a single character you can add curly brackets, `_{...}`.

`x_i`	x_i
`x_{i+1}`	x_{i+1}

Superscripts are the same except the command is `^` and if your superscript is multiple characters, `^{...}`.

`x^2`	x^2
`e^{x+1}`	e^{x+1}

You can also combine subscripts and subscripts,

`x_i^2`	x_i^2
`e_{i+1}^{x+1}`	e_{i+1}^{x+1}
`e_{i+1}^{x^2}`	$e_{i+1}^{x^2}$

Square roots and nth roots are created using the `\sqrt{...}` command,

`\sqrt{x+y}`	$\sqrt{x+y}$
`\sqrt[3]{x+y}`	$\sqrt[3]{x+y}$

Sums are made using `\sum{...}` and products are made using `\prod{...}`.

`\sum_{n=1}^{N} n^2` produces,

$$\sum_{n=1}^{N} n^2 \tag{5}$$

and `\prod_{n=1}^{N} n^2` produces,

$$\prod_{n=1}^{N} n^2 \tag{6}$$

Integrals are very similar to sums and products, `\int_{a}^{b} x^2 dx` results in,

$$\int_a^b x^2 dx \tag{7}$$

If you want to place the limits of integration on top of the integral instead of next to it, you can use `\int \limits_{a}^{b} x^2 dx`,

$$\int\limits_{a}^{b} x^2 dx \tag{8}$$

Limits can be written by using `\lim_{...}` command. `\lim_{x\to\infty} f(x)` produces,

$$\lim_{x\to\infty} f(x) \tag{9}$$

One of the things you might have noticed is that characters used in an equation are a different font than normal text. This means that if your equation has a cosine in it, if you where to just type out `cos(\theta)`, your cosine would look like $cos(\theta)$. However, we want our cosine to use the standard font, so we use the cosine command `\cos`.

`\sin(\theta)`	$\sin(\theta)$	`\arcsin(\theta)`	$\arcsin(\theta)$
`\cos(\theta)`	$\cos(\theta)$	`\arccos(\theta)`	$\arccos(\theta)$
`\tan(\theta)`	$\tan(\theta)$	`\arctan(\theta)`	$\arctan(\theta)$

6 Figures

To include figures in your paper, you need the graphicx package, `\usepackage{graphicx}`. Inserting a figure is similar to inserting an equation, where you have a begin and end command. Instead of passing "equation" as an argument, you pass "figure".

```
\begin{figure}
...
\end{figure}
```

Any commands pertaining to the figure goes in between the begin and end commands. To add a figure, you use the `\includegraphics[options]{figurename}` command, and pass the filename to the command. If you don't include a file path to your figure, then the image must be in the same location as your .tex file. One of the options you can pass is the desired size of the figure. There are width and height options, if only one of them is set, then the aspect ratio of the figure is maintained. Here is the code used in inserting Figure 1 earlier in the tutorial.

```
\begin{figure}[b]
\centering
\includegraphics[width=0.9\textwidth]{env.png}
\caption{Example of the TeXShop environment.}
\label{env}
\end{figure}
```

In the above example, the width of the figure is set to `[width=0.9\textwidth]`, which means that the figure will be rescaled so that it will take up 90% of the width of the paper, excluding the margins. If you don't set a width or height of the figure, the image will be inserted as the actual size of the image. The `\centering` command centers the figure in the paper, and the `[b]` option

in the begin command tells LaTeX to place the figure at the bottom of the page. A note about the placement of figures, LaTeX places the figure automatically on the page it feels works best. Adding a caption to the figure is as easy as `\caption{...}` and labeling works the same as with equations.

7 Tables

Inserting tables is just like inserting figures, but we now pass the argument "tabular",

```
\begin{tabular}{c c | c c}
...
\end{tabular}
```

The other option that is passed in the above example is the number of columns in the table. In this case there are four columns, center justified and a line between the second and third columns. Each "c" represents a center justified column and the pipe, "|" adds a line between columns. You can also make a column left justified by using an "l" or right justified using an "r" instead of "c", but be careful not to confuse the pipe symbol "|" and the letter "l". An example table can be made using,

```
\begin{tabular}{c r | l c}
1 & 2 & 3 & 4 \\
a & b & c & d \\
\hline
$x$ & $y^2$ & $\rho$ & $\cos(\theta)$
\end{tabular}
```

which produces,

1	2	3	4
a	b	c	d
x	y^2	ρ	$\cos(\theta)$

Each line after the begin command creates a new row in the table. Each element of the row is separated by an &. After the last element, to create a new row you us a double backslash `\\`. If you want a horizontal line, you use the `\hline` command. Finally, the table can contain whatever you like, such as text, numbers, and equations, and the width of the columns will automatically adjust to fit.

8 Templates

LaTeX has the ability to use templates so you don't have to set up all the formatting yourself. This section will focus on two templates, the American Meteorological Society template and the TAMU Thesis Office template, as well as how to use references in your paper.

8.1 AMS Template

For submissions to any of the AMS journals, like Monthly Weather Review (MWR) or the Journal of Atmospheric Sciences (JAS), you have to use the AMS template. It can be downloaded from their website at,

https://www2.ametsoc.org/ams/index.cfm/publications/authors/journal-and-bams-authors/author-resources/latex-author-info/

or from my github repository at https://github.com/mherr77m/LaTeX.

The directory that you download contains a bunch of files, but the important ones that you need for your paper are ametsoc.cls, ametsoc2014.bst, template.tex, and references.bib. The template.tex file is where you will write your paper and references.bib is where you put your references. The other two files are used for formatting and you shouldn't need to edit them.

Let's start with the template.tex file which you can open using TexShop or your desired LaTeX front end. The first thing that you'll notice is all of the comments. You can leave comments in your code by using `%`. The first actual command is defining the document class, which is now defined as ametsoc. This is where you tell LaTeX that you want to use the AMS template. The next command `\journal{...}` lets you define which journal you are submitting to, since they don't all use the exact same formatting. Lines 53 through 91 help you set up your title page. The last part of the preamble is the abstract and all you have to do is put the text of your abstract within the `\abstract{...}` command.

The main body of the paper is where you will write your paper using all of the tools discussed earlier in this tutorial. The template also has areas for acknowledgments and appendixes. AMS journals request that you put all the tables and figure at the end of the paper, so from line 186 to the end is where you would place your tables and figures.

8.1.1 References

A very important thing you need to do to make sure your references work in the template is to uncomment lines 182 and 183, `\bibliographystyle{ametsoc2014}` and `\bibliography{references}`. These two lines establish the formatting for the bibliography and then finds your references. In the template directory, you references will be put into references.bib.

When you open references.bib in TexShop, the top half of the file contains a long list of journal name abbreviations that you can use when defining your references. At the bottom are sample references that you can delete and replace with your own references. For this tutorial, I'm only going to discuss one type of reference, a journal article. Here is an example reference definition,

```
@article{Lorenz1969,
      author = {E. N. Lorenz},
      title = {Predictability of a flow which possesses many scales of motion},
      journal = TELLUS,
      year = {1969},
      volume = 21,
      pages = {289--307}}
```

The first line is the type of reference, in this case a journal article, which you pass several arguments describing the article. The first argument, `Lorenz1969b` is the label for the reference.

When you want to use this reference in the paper, you will use this label. The next two lines are for the authors and title. The journal argument is where you can use the abbreviations from above, in this case it's not really an abbreviation since the journal name is short. The final three lines are self explanatory, they are for year, volume, and page numbers of the journal article.

You add all of your references like this to the bottom of references.bib. The way that referencing works in LaTeX is that when you compile your paper, the program looks through your references.bib file for the label and then puts the reference into your paper. This means that your references don't have to be in any specific order and you can even have references that you don't end up using in the paper.

Using the AMS template, there are two commands you can use in your paper to add citations, either `\citet[...]{...}` or `\citep[...]{...}`. Here are some sample citation commands and how they will appear in the paper,

`\citet{jon90}`	Jones et al. (1990)
`\citet[chap. 2]{jon90}`	Jones et al. (1990, chap. 2)
`\citep{jon90}`	(Jones et al. 1990)
`\citep[chap. 2]{jon90}`	(Jones et al. 1990, chap. 2)
`\citep[see][]{jon90}`	(see Jones et al. 1990)
`\citep[see][chap. 2]{jon90}`	(see Jones et al. 1990, chap. 2)
`\citet{jon90,jam91}`	Jones et al. (1990); James et al. (1991)
`\citep{jon90,jam91}`	(Jones et al. 1990; James et al. 1991)
`\citep{jon90,jon91}`	(Jones et al. 1990,1991)

The final part of using citations is how to compile LaTeX with references. If you are using TexShop, you'll first have to "Typeset" using LaTeX from the drop down menu seen in Figure 1. Once that is finished compiling, you'll change the LaTeX option to "BibTex" and then typeset/compile again. This compilation should be very quick, then you'll chose LaTeX from the drop down menu once again and compile two more times. These are all the basics for using the AMS template.

8.2 Thesis Template

The TAMU thesis template is a bit different from the AMS template, and can be found either in my github repository, or from

http://howdy.me

A large difference with this template is that it is much more modular than the AMS template. You'll notice that there are several .tex files, luckily they are all named so that you know what each file is used for. The main .tex file is tamuthesis.tex, which is the file you will compile. You will not have to modify this file very often. The first changes that need to be made start on line 55 and goes through line 68. This is where you define your title, author, committee, etc. After you define your title page, there are several `\include` commands which are used to call the other .tex files. For example, you will write your abstract in abstract.tex, then tamuthesis.tex will compile the abstract and add it to the main paper. The only other thing you should need to change is to include additional sections as you need them, since only three are included in this template.

The other .tex files are where you will actually write your paper. Each .tex file already has some example text in them, so it is fairly straight forward on how to edit them and add your text. One thing to notice is that figures are now placed in a "figures" directory, so you'll need to make sure to include the directory name when inserting the figures. For an example of this, see line 34 of section1.tex.

References in this template are similar to the AMS template. There is a references.bib file which contains lots of sample references. You can add your references to this file and then cite them in your paper just like you would for the AMS template. When you are ready to compile, you will compile tamuthesis.tex and you will compile it the same was as for an AMS paper.

9 Wrap Up

This tutorial only covered the basics of how to use LaTeX to write your journal article or thesis. There are many resources for any other questions you have. If you have a problem in LaTeX there is a good chance someone else has had the same problem and has solved it. Here are some websites to check out for more help.

https://en.wikibooks.org/wiki/LaTeX
https://www.sharelatex.com/learn/Main_Page
http://tex.stackexchange.com/help
https://latex-project.org/ftp.html