

LaTeX Tutorial

Boris Burkov
Costandino Dufort Moraites
Gautam Kamath

1 Introduction

I remember when I first started hearing about Latex, I thought it was something new, but it is important to keep in mind that Latex has been around since the 80s! When thinking about picking up any new piece of technology, it's easy to be hesitant because you aren't sure if it will stick or be general enough to be worth your time learning it. Latex has already stuck and it is extremely general. You will not regret learning how to use it if you spend any significant amount of your time preparing technical documents.

Despite seeming daunting at times, there is nothing to be afraid about switching over to Latex for writing technical documents. It instantly makes your writing immortal and forever editable which is advantageous over paper mediums which inevitably get lost or misplaced, and which become difficult to edit without rewriting from scratch. Latex is designed for typesetting beautiful technical documents, a feature the equation editor in Microsoft Word or NotePad cannot boast. With Latex your writing starts off looking professional which makes the job of whoever will be reading your work much easier.

The first two sections will cover enough of the basics to get you to write your first document in Latex from scratch. From there, there are additional topics for more complicated things you may want to do when writing something. The basics include the general Section, subsection model, how to enter equations, and how to turn your tex file into a pdf.

Finally, I will say if after the next two sections you still feel intimidated by Latex, try using Lyx. Lyx is a GUI from Latex which abstracts away all of the TeX syntax. Lyx is available on all of the Windows, Mac, and Linux platforms. There is a short tutorial which comes with Lyx which will get you on your feet in about 30 minutes, and will help you to understand how to use Latex without being confused by the syntax. Throughout the first few sections I will scatter tangential remarks about Lyx but for the most part it works like any other GUI, it gives you menus to navigate through so you can figure out how to tell Lyx to give you a Latex document how you want it without worrying about the Latex markup language.

Oh! And this document is not intended to be all encompassing! Google is your friend!

2 Installing Latex

2.1 Testing Your Installation and What You'll Need

In general the installation methods described below will install a lot of additional software including editors specifically for LaTeX, but for the sake of simplicity this tutorial will focus on methods consistent

across all platforms. After following the platform specific instructions for installation, test your installation by checking to see if you have pdflatex installed. This means opening a terminal or command prompt and typing the following (platform independent):

```
> pdflatex
```

Afterwards you should get output asking about the name of your input file or really just any output which isn't an error saying that pdflatex couldn't be found. pdflatex is the command you will use to convert your .tex files to .pdf files and it's all you'll need for most documents.

2.2 Linux

Installing everything you need to get started in a Linux based OS is extremely simple. The preferred package is texlive.

```
>sudo apt-get install texlive
```

If you want the full texlive packages use (it might take a while to download):

```
>sudo apt-get install texlive-full
```

At least for the basic tutorial, you won't need anything in the full package. Also, if you later find that you want additional functionality, you can simply install new packages, which is covered later.

2.3 Windows

In windows the preferred Latex packages are from MikTeX.

- (i) Download the packages/installer here: <http://www.miktex.org/2.9/setup> (I'll assume you got the full packages, and be warned it will take a while to download.)
- (ii) From here run the installer.

2.4 Mac

The easiest method from this link: http://guides.macrumors.com/Installing_LaTeX_on_a_Mac is summarized below. For other more advanced methods see the link. Similarly, the simplest method is included here for completeness, but the address linked is a living document and more likely to be up to date.

Installing latex on Mac the easy way:

- (i) Download the MacTeX Package here: <http://mirror.ctan.org/systems/mac/mactex/MacTeX.mpkg.zip>
- (ii) Unzip the downloaded file and run the .mpkg file to install latex.

2.5 Lyx

Lyx is a GUI on top of Latex which behaves like traditional word processors. It abstracts away the Latex syntax and provides the user menus to help figure out what they want. If you are frustrated with straight Latex, give Lyx a try. It will help you see what is possible with Latex and has a somewhat more friendly learning curve. Lyx is available for all platforms mentioned in this guide.

3 Hello World : The Basics

Now we are going to go through the process of creating a simple latex document which will encompass all of the basic features you'll need to get by.

First create a new tex file. This simply means opening your favorite text editor (vim, emacs, notepad++, notepad, nano, gedit, etc) and save a file with a .tex extension. Be careful if you are using windows and you haven't enabled the display of file extensions in the explorer window manager. Doing this will make it easier for you to work with .tex files.

I'm using Ubuntu and I compose my tex in vim so to start my new document I simply do:

```
> touch example.tex
> vim example.tex
```

Below is a sample document with the basics. I recommend perusing it a bit, then going through the section on the bottom and typing the relevant LaTeX in to your example.tex so you can get a feel for how straightforward LaTeX syntax really is.

```
% comments in LaTeX start with a %
% This is my preamble, it sets up my margins and includes any packages I'll need
% Both will be discussed in more detail later.
\documentclass[12pt]{article}

% (1) include the extra packages I'll need
\usepackage{amstext, amsmath, amssymb}

% (2) set up my borders
\addtolength{\oddsidemargin}{-1.0in}
\addtolength{\evensidemargin}{-.875in}
\addtolength{\textwidth}{1.75in}
\addtolength{\topmargin}{-.875in}
\addtolength{\textheight}{1.75in}

%(3) every document begins with \begin document and ends with
%\end{document}
\begin{document}

%(4)
\title{Hello World! \\ The Basics }
\author{Cadmium \\ Cadmium } %terminate lines with \\
\date{} %this tells LaTeX to use today's date
\maketitle %this tells LaTeX to format the code above as the title of the
document

\section{The definition of Convergence!} %(5) This is how you declare a new
section
```

We will start with a section on the definition of convergence. Then a section discussing its properties.

`\subsection{Definition}` (6) This is how you give a section subsections
The n th partial sum, S_n of a sequence $\{a_n\}$ is the sum
of the first n terms of the sequence. (7) so this is an example of inline
equations

A series $\{a_n\}$ is said to be convergent when the sequence of
its partial sums, $\{S_1, S_2, \dots\}$ converges. A series converges
if there exists a limit L such that for any $\epsilon > 0$, there
exists N such that for all $n \geq N$.

`\begin{equation}` (8) and here is a sample equation broken out from the text
 $|S_n - L| \leq \epsilon$
`\end{equation}`

`\subsection{Reflection}` (9)
Note the relationship here to compactness. Every infinite sequence
in a compact space is convergent, ie taking an infinite number of steps in a
compact space
means most of the steps need to be relatively close to each other!

`\end{document}` (10)

3.1 Borders and Packages (1) and (2)

Borders and packages will be covered more later. Right now just see that there is a sample of something
you can play with if you don't like my margins and the packages I included are the ones I tend to use
for typesetting my equations. Feel free to use the same packages or look around for others which you
think suit your needs more.

3.2 Beginning and Ending a Document (3) and (10)

The text of a document begins with

```
\begin{document}
```

and ends with

```
\end{document}
```

Make sure you have these or latex will bark at you.

3.3 Titles (4)

Notice how you can use `\\` to terminate lines and how specifying date with no date will pick up today's
date.

3.4 (5) and (6) and (9)

Unlike HTML, you don't have to close nodes. They close themselves in reasonable ways. Keep in mind there are also subsubsections!

3.5 (7) and (8)

Here is probably the most useful part, and certainly the most liberating part if you are used to using various other equation editors. There are two main ways of inserting equations in LaTeX: by enclosing things with dollar signs for in-line equations and by enclosing equations with the equation environment (`\begin{equation} ... \end{equation}`) for equations broken out from the text.

You get exponents with a caret (^) and subscripts with an underscore (-) for arguments to the exponent or subscript, be sure to enclose them in { and }! You insert special symbols with `\SYMBOLNAME` for example `\epsilon` (ϵ), `\gamma` (γ), `\geq` (\geq), and `\in` (\in).

Alright. That should be enough to get you started. Give it a shot. It really isn't so bad and the resulting beautifully typeset documents are worth it.

4 Tables

Suppose you want to write up your multiplication tables in LaTeX, so that you will never forget them, or maybe you need to type up a truth table for a logic problem. Either way, LaTeX comes with a powerful environment called tabular for creating tables. To make a table using tabular, we do something like the following simple example, the details of which I will explain after you take a look at it:

```
\begin{tabular} {| l | c || r |}  
  \hline  
  1 & 2 & 3 \\ \hline  
  4 & 5 & 6 \\ \hline  
  7 & 8 & 9 \\ \hline  
\end{tabular}
```

which yields:

1	2	3
4	5	6
7	8	9

Let's break down the different parts of this syntax and discuss what they do. The line `\begin{tabular} {| l | c || r |}` puts us into the tabular environment, and declares that we will be creating a table with 3 columns. One that is left justified, one that is center justified and one that is right justified. The vertical lines between and around the columns indicate what kind of borders we want the table to have between columns. In this case, we put a double line between the 2nd and 3rd columns and a single line between each other column and on the outside of the table. If we wanted to have more columns, we would just have to specify more column items here. For instance, if we were to have written: `\begin{tabular} {l c c c c r}` here, that would give us a table with 6 columns and no lines acting as borders between any of them. The remaining 5 lines:

```
\hline
```

```

1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
7 & 8 & 9 \\ \hline
\end{tabular}

```

fill out the data in the table. The `\hlines` put a horizontal line at the location we are currently at in the table, the numbers fill out the columns of the table and the `&` and `\\` serve as column and row delimiters, respectively.

We could just as easily put words instead of numbers into the table, for instance if we wanted to label the columns and rows. If we wanted to put entire paragraphs, we would replace the column names like `l`, `c`, or `r` with `p{width}`. Where `width` denotes the maximum width of the paragraph column in some measurement like inches or centimeters.

5 Images

Imagine that we are writing something, and would like to include an image, such as a graph or diagram. LaTeX allows us to do this quite easily as well. We need to include a package called `graphicx` which should come bundled with your installation of LaTeX. Once that is included, all we needs to do to include a picture into our document is to use the line:

```
\includegraphics{PATH_TO_IMAGE}
```

where we would like the image. The path can be either a relative path starting from where the `.tex` file defining the document is or an absolute path. We can also provide options in square brackets to scale the image:

```
\includegraphics[scale=0.75]{PATH_TO_IMAGE}
```

This will render the image scaled uniformly by a factor of 0.75. Other options include translation, rotation, cropping, and much more.

6 Packages

A few short notes on installing packages. In general, installing packages can be complicated, but we focus on the simplest case where the package simply consists of a few `.sty` files.

In this case, installing a new package, such as the algorithmic package for including really snazzy pseudocode in your write-ups is a two step process:

1. Find the appropriate `.sty` files (in this case `algorithm.sty` and `algorithmic.sty`).
2. Run `texhash` on the directoy where you want to install the packages. In my case in `/texmf`, where I put the `.sty` files in `/texmf/tex/latex`.

Now we discuss each step in a little more detail. For the algorithmic example, you can download the relevant `.sty` files here: <http://texcatalogue.sarovar.org/entries/algorithms.html>

And after downloading and unzipping the files, you need to place them in the correct directory. I use Ubuntu and on Ubuntu it is common to install new packages either in the packages directory of my `texlive` install, or in the directoy `/texmf/tex/latex/PACKAGENAME`. Note that

if this is your first time installing a package you may have to create the directories texmf, tex and latex. If you are on Windows, this directory will be something like ProgramFiles/MikeTex 2.7/texmf-local/tex/latex/PACKAGENAME. In general, your latex install should add the command 'texhash' to path so if you are having trouble finding where to put your newly downloaded packages, run texhash from a terminal with no arguments. In the output of texhash, you will see texhash looking in some directories for new packages, and that is where you should put the new packages you want to install.

If you are having still trouble finding this directory, you can go here for the full details on package installation: http://en.wikibooks.org/wiki/LaTeX/Packages/Installing_Extra_Packages

In any case, let's say I put the files algorithmic.sty and algorithm.sty into /texmf/tex/latex/algorithm. Now all I have to do is run the command:

```
> texhash ~/texmf
```

This installs the algorithms environment and now I can use the algorithms package with a

```
\usepackage{algorithm, algorithmic}
```

With this use package statement,

```
\begin{algorithmic}
\IF {$i\geq maxval$}
    \STATE $i\gets 0$
\ELSE
    \IF {$i+k\leq maxval$}
        \STATE $i\gets i+k$
    \ENDIF
\ENDIF
\end{algorithmic}
```

Will look like this:

```
if  $i \geq maxval$  then
     $i \leftarrow 0$ 
else
    if  $i + k \leq maxval$  then
         $i \leftarrow i + k$ 
    end if
end if
```

For more help with the algorithmic environment, see here: http://en.wikibooks.org/wiki/LaTeX/Algorithms_a

7 Bibliographies

Now in writing a paper or problem set writeup, you may often find yourself wanting to cite a source that you have been using. The simplest way to do this is to use the thebibliography environment. To generate the very simple bibliography that follows, we use the following LaTeX:

```
\begin{thebibliography}{9}
\bibitem{mac_install}
```

```
    http://guides.macrumors.com/Installing\_LaTeX\_on\_a\_Mac
\bbitem{ubuntu}
    https://help.ubuntu.com/community/LaTeX
\bbitem{example}
    Boris, Constantin, Gautam 2011: A simple dummy bibliography entry
\end{thebibliography}
```

Each `bbitem` corresponds to a new entry in your bibliography, and the argument passed in to `bbitem` is called a cite key. This allows you to later just throw in `\cite{CITE_KEY}` to add a citation wherever you want, which I will do at the end of this sentence. [3]

References

- [1] http://guides.macrumors.com/Installing_LaTeX_on_a_Mac
- [2] <https://help.ubuntu.com/community/LaTeX>
- [3] Boris, Constantin, Gautam 2011: A simple dummy bibliography entry