

Brief Introduction to L^AT_EX

Prepared by Anna Walsdorff
anna.walsdorff@rochester.edu

Updated: September 14, 2018

License

This document is released under the Creative Commons Attribution license () <http://creativecommons.org/licenses/by/3.0/>.

The source code for this document is available on Github at <https://github.com/ffrodslaw/latex-intro>.

This document incorporates material from previous L^AT_EX courses taught by [Chitrlekha Basu](#) and [Casey Crisman-Cox](#).

Contents

I	Introduction and Basics	1
1	What is \LaTeX ?	1
2	Installation	1
3	First document	2
4	Special characters	2
5	Basic tweaks	3
6	List Environments	4
7	Getting help	6
II	Tables, figures, footnotes, etc.	7
8	Tables	7
9	Figures	9
10	Footnotes and endnotes	10
11	Abstract and titles	10
12	Sections	11
13	Advanced tweaks	11
III	Typing math	13
14	Basic math	13
15	Fancy equations	15
16	Matrices	16
17	Theorems and proofs	17
18	Fancy fonts	17
IV	Presentations with beamer	18
19	Theme and colors	18
20	Frames	19
21	Appendix slides	21
V	Managing citations with $\text{Bib}\TeX$	22
22	Manually adding references	22
23	Bibliography management	22
24	\LaTeX with $\text{Bib}\TeX$	23

I Introduction and Basics

1 What is L^AT_EX?

- L^AT_EX (or LaTeX) is pronounced “lay-teck”, not “latex”
- L^AT_EX is a standard open-source, document preparation system used for any kind of technical or academic writing. It is especially useful for typesetting math.
- In contrast to Microsoft Word, L^AT_EX is not WYSIWYG (what you see is what you get), but WYSIWYM (what you see is what you mean). What that means will become a bit clearer once we write our first document.
- Files are all text and PDF. There are no compatibility issues across operating systems or version. L^AT_EX is available for Linux, Mac, and Windows. For free!
- You can prepare all kinds of documents with it: articles, presentation slides, books, posters, CVs, letters, lecture notes (if you’re so inclined.)
- Depending on your handwriting, using L^AT_EX will make your TAs very happy.

2 Installation

Please make sure you have L^AT_EX installed on your personal machine.

Linux Use your software manager or terminal to install T_EXLive. For example, on Ubuntu use the command

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

Mac On a Mac OS X machine you’ll want to use MacT_EX. You can download the installer at <http://www.tug.org/mactex/>.

Windows On Windows the standard is MiK_TE_X. Download it from <http://www.miktex.org/download>.

In addition to the L^AT_EX engine, you will want an editor for it. There are dozens of options and the choice is often very personal. Some editors that people seem to like are T_EXstudio, T_EXmaker, Sublime Text, Emacs, or Atom. Choose one you like and download the version for your operating system. I will be using T_EXmaker which you can download at <http://www.xmlmath.net/texmaker/download.html>.

3 First document

Now let's write our first ever \LaTeX document. Open up your editor of choice and type the following.

```
\documentclass{article} %tell it what kind of thing we're writing.
%Use article unless you're doing something fancy
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
``And now the time has come,'' the Walrus said, ``to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.''

This is a new paragraph
\end{document}
```

Before creating our output document, let's save the code we just wrote as "Test.tex". tex is the extension of all text files written in \TeX . Once we have saved the file, we can compile it (process it with the \LaTeX engine) by clicking Tools > Build (or similar depending on the editor).

A few things to note about our document:

- First, % is how we generate comments. These are remarks that we want to leave for ourselves or others if the source file, but that we don't want to show up in the actual document. You can imagine having part of a document that you're not sure if you want to include. Rather than move it or delete it, you can simply comment it away and reinstate it as needed.
- Another thing to note is that the only text to appear in the document is what is between the commands begin and end document. This is no coincidence. Everything above `\begin{document}` is about setting up the document. In this case it just means telling it we want to write an article, but we'll add things here as we go. Everything below `\end{document}` is just ignored by the compiler. This another place to hide text you're not sure if you want to include.
- Looking back at our example you'll also notice that white space doesn't matter much. Extra spaces and line breaks do not have an effect on the output. This allows you some liberty to structure your text in the source file in a way that is most readable to you.
- Another thing to notice is the use of quotation marks. This is something that often confuses novice users. The double quote mark " (shift-') produces quotes that only curve one direction (same for single quotes from that key).

4 Special characters

For writing your .tex file you will only include the characters on your keyboard. Every special character such as

The following characters are special: % # \$ ^ & _ { } ~ \. They mean something specific to \LaTeX , so if you type for example "%" in your text file, you won't see it in the output. All \LaTeX commands start

with a `\`. Most of the special characters can be displayed by attaching a backslash to them. So in order to type a percentage sign, you need to type `\%` in your `.tex` file.

5 Basic tweaks

By default \LaTeX uses single spacing, 10pt font, and rather large margins. If you want your document to look differently, it's easy to change these things.

Font size

To change to change the font size in your document, simply alter the first line of our code in which we define the document class.

```
\documentclass[11]{article}
```

Margins

To adjust the margins we will use our first package. Packages are add-ons to \LaTeX for increased functionality. There are hundreds of them for a variety of uses. For margins we will use the `geometry` package.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
```

Note that we include packages before the start of the “actual” document, in the *header* of our `tex` file.

Spacing

Double or 1.5 spacing often can make papers more readable. We can use the `setspace` package to adjust our header.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing %for 1.5 spacing use \onehalfspacing
```

Other notable commands for spacing are:

- Two or more carriage returns (enter keys), `\\`, and `\linebreak` will move all following text to a new line. A single carriage return will produce a non-breaking space.
- The special character `~` also produces a non-breaking space.
- `\par` will create a new paragraph for the following text.
- `\linebreak` and `\pagebreak` move all following text to the next line and page respectively.
- Any number of consecutive spaces are treated as a single space.

6 List Environments

You can create lists of items, numbering, or descriptions.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
``And now the time as come,' the Walrus said, ``to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.'

This is a new paragraph. Here is a list of animals
\begin{itemize}
  \item Ant
  \item Elephant
  \item Lizard
\end{itemize}
\end{document}
```

It's easy to do it with numbers instead. Simply replace `itemize` with `enumerate`. Remember to do this at the beginning and at the end of the environment. The `\item` command remains the same.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is fun.
``And now the time as come,' the Walrus said, ``to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.'

This is a new paragraph. Here is a list of animals
\begin{enumerate}
  \item Ant
  \item Elephant
  \item Lizard
\end{enumerate}
\end{document}
```

We can nest lists and alternate between them:

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing
\begin{document}
```

```

This is my first document using \LaTeX.
\LaTeX\ is                fun.
``And now the time as come,'' the Walrus said, ``to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.''

```

This is a new paragraph. Here is a list of animals

```

\begin{enumerate}
  \item Ant
  \item Elephant
  \begin{itemize}
    \item Asian Elephant
      \begin{enumerate}
        \item Sri Lankan elephant
      \end{enumerate}
    \item African Elephant
    \item Mammoth
  \end{itemize}
  \item Lizard
\end{enumerate}
\end{document}

```

The description environment is a bit different as it requires labels which replace the bullets or numbers.

```

\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing
\begin{document}
This is my first document using \LaTeX.
\LaTeX\ is                fun.
``And now the time as come,'' the Walrus said, ``to talk of many things.
Of shoe and ships and sealing wax, of cabbages and kings.''

```

This is a new paragraph. Here is a list of animals

```

\begin{description}
  \item[Ant]
  \item[Elephant] Some elephants include
  \begin{itemize}
    \item Asian Elephant
      \begin{enumerate}
        \item Sri Lankan elephant
      \end{enumerate}
    \item African Elephant
    \item Mammoth
  \end{itemize}
  \item[Lizard]
\end{description}
\end{document}

```


7 Getting help

Especially in the beginning (but even if you're a pro!), you will run into problems in your code. You either don't know how to do a specific thing in \LaTeX or you get an error message when your code won't compile. Google is your friend! Searching for "how to do [xyz] in latex" or searching for a meaningful portion of the error message will help you in nearly all cases. Some websites that are especially helpful are:

- <http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>. This PDF is a very comprehensive lists of symbols and how they can be reproduced in \LaTeX - Note that all of them require you to load a special package.
- <http://detexify.kirelabs.org/classify.html> is a website where you can draw a symbol that you're trying to reproduce and it tries to find it. There are also Android/iOS apps that work similarly.
- <https://en.wikibooks.org/wiki/LaTeX> is a good source with examples for basic things.

Feel free to also ask me for help if you can't find it on the internet. My general advice is to comment out (with %) large portions of the text if you run into compiling issues. If your document compiled before, comment out all changes you have made since then and gradually introduce your previous changes until you find the line of code that prevents \LaTeX from compiling. The default error messages may give you the line at which it fails but often they are not very useful.

II Tables, figures, footnotes, etc.

Figures and tables are especially important for writing academic papers.

8 Tables

Let's start with a new document.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing
\begin{document}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on this
topic.
\end{document}
```

Let's add a table.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing
\begin{document}
This paper examines important scientific questions.
It does so in a super rigorous and presents results so well-identified that
they can't grab a bite without being spotted.
We address all the important concerns with our results and can say with
absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal
the paper is.
This paper advances all of science by about 25 years.
In fact, we have produced 200\% more science than anyone else writing on this
topic.

\begin{table}[h]
\centering
\caption{Table of stuff}
```

```

\label{tab:Stuff}
\begin{tabular}{rc|l}
\hline
Input 1 & Input 2 & & Sum \\ \hline
5 & 10 & & 15 \\
2 & 9 & & 11 \\ \hline
\end{tabular}
\end{table}

Table \ref{tab:Stuff} shows just all the cool science we do.
\end{document}

```

What does it all mean?

- The entire table is between the `\begin{table} ... \end{table}` commands. The `[h]` tells it we want the table placed “here.” This is just a suggestion as the software will place it where it thinks is best. Other options include `b` for bottom, `t` for top, and `p` for separate page. We use the centering command to center the table. I then gave it a caption and a label. You should always give your tables a caption because that tells the reader what’s going on in the table. The label is for you to use to refer to the table within your document. We use the `\ref` command to refer to the table in the text. This means that we never have to manually number or adjust the numbers. Quick note is that we can move caption and label to below the tabular commands and this moves the caption to below the table, but label has to come after caption.
- After this we get to the tabular section. This is where the actual table happens. The first thing after we open with `\begin{tabular}` is that we specify `rc|l`. This tells the software two things: First, it tells it there will be three columns (number of letters is the number of columns). The second thing is how to justify the text within each column with `r`, `c`, and `l` meaning right, center, and left, respectively. The pipe, `|` says to put a vertical line between these two columns. We use `\hline` to create horizontal lines after a row, `\\` to end rows, and `&` to separate columns. Everything else is just the text.

Something that often comes in handy is merging cells together. We can do that with the `multicolumn` command. It takes three arguments. The first is how many columns to take up, the second is the alignment, and the third is the text.

```

[...]

\begin{table}[h]
\centering
\caption{Table of stuff}
\label{tab:Stuff}
\begin{tabular}{rc|l}
\hline
Input 1 & Input 2 & & Sum \\ \hline
5 & 10 & & 15 \\

```

```

                2                & 9                & 11 \\ \hline
                \multicolumn{3}{1}{here's a remark.}
            \end{tabular}
\end{table}

[...]
```

9 Figures

Incorporating figures is very similar to tables. We are going to use the `graphicx` package.

```

\documentclass [11]{article}
\usepackage [top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing
\usepackage{graphicx}

[...]
```

Table `\ref{tab:Stuff}` shows just all the cool science we do. `\ref{fig:plots}` graphs some other stuff.

```

\begin{figure}[h]
\centering
\includegraphics [width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}
\label{fig:plots}
\end{figure}

[...]
```

- The external files you import will have to be either `.jpg`, `.png`, or `.pdf`. Place the files in the same directory as your `.txt` file.
- Figures start the same as tables with a `begin` and an `end` and they use the same placement indicators. To actually include the figure we use the command `\includegraphics [options]{name}`. In the options we tell the software to scale the picture. We can scale it by including `width=0.5\columnwidth` to get it to take half the page. Or we can scale the whole image by a fraction by using `scale=.9` as in `\includegraphics [scale=.9]{name}`.
- The file of the image goes in the brackets. It needs to be the fixed or relative path to the image. There should be no spaces in the name of the image either. Like with tables we caption and label to identify them.

- The `\centering` command and the `figure` environments are optional. The `figure` environment is another float (like the `table` environment). It creates a “container” for your image, and again, allows you to title and number the image using the `\caption{}` command.

10 Footnotes and endnotes

We often may want to include things that are important but may interrupt the flow of the text. That’s when we use footnotes.

```
[...]
```

```
That figure sure was a pretty.\footnote{The prettiness of the figure was rated
    as 7.9 on the princess beauty scale.}
```

```
[...]
```

If you instead want end notes, you can simply use `\endnote{}` instead. Note that you need to load the `endnotes` package as well.

11 Abstract and titles

Every academic paper you write should have an abstract and a title page. The actual title will be printed by the command `\maketitle`. However, we need to supply \LaTeX with information on the title, author, and date.

```
\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{graphicx}
\usepackage{setspace}
\onehalfspacing

% % New % %
\title{On the Making of Science}
\author{Bullwinkle J. Moose\thanks{Ph.D. Candidate, Moose Institute for
    Toetapping.}}
\date{\today}
% % % % %

\begin{document}

\maketitle
\newpage

[...]
```

To add an abstract, we simply can use the `abstract` environment before the page break.

```
[...]

\maketitle
\begin{abstract}
This paper does stuff
\end{abstract}

\newpage

[...]
```

12 Sections

Each document class comes with a built-in hierarchy and set of sub-headings. The `article` class allows the following sub-headings, in the following order: `\part{}`, `\section{}`, `\subsection{}`, `\subsubsection{}`, `\paragraph{}`, `\subparagraph{}`. If an “*” is placed after the subheading name, such as `\part*{}`, then the heading will not be numbered.

13 Advanced tweaks

Font styles and sizes

To change the font style or size you either use an the environment approach or the command approach. By the first approach, the text contained within the environment will be in a different font style or size. By the second approach, only the text included as the mandatory argument to the command will be altered. Examples of font styles are **bold face** and `SMALL CAPS`. How to express the most common font styles is summarized below.

Table 1: Font styles

Style	Environment	Command
normal/default	<code>\begin{textnormal} ... \end{textnormal}</code>	<code>\textnormal{...}</code>
bold	<code>\begin{bf} ... \end{bf}</code>	<code>\textbf{...}</code>
<i>italics</i>	<code>\begin{it} ... \end{it}</code>	<code>\textit{...}</code>
<code>SMALLCAPS</code>	<code>\begin{sc} ... \end{sc}</code>	<code>\textsc{...}</code>
<u>underline</u>	none :-)	<code>\underline{...}</code>
Roman	<code>\begin{textrm} ... \end{textrm}</code>	<code>\textrm{...}</code>
Sans Serif	<code>\begin{textsf} ... \end{textsf}</code>	<code>\textsf{...}</code>
teletype	<code>\begin{tt} ... \end{tt}</code>	<code>\texttt{...}</code>

Similarly, you can use change font size with 10 different declarations. Note that these sizes are always relative to the default font size that you specified in the header.

Table 2: Font sizes

Style	Environment	Command
<small>tiny</small>	<code>\begin{tiny} ...\end{tiny}</code>	<code>\tiny{...}</code>
<small>scriptsize</small>	<code>\begin{scriptsize} ...\end{scriptsize}</code>	<code>\scriptsize{...}</code>
<small>footnotesize</small>	<code>\begin{footnotesize} ...\end{footnotesize}</code>	<code>\footnotesize{...}</code>
<small>small</small>	<code>\begin{small} ...\end{small}</code>	<code>\small{...}</code>
<small>normalsize</small>	<code>\begin{normalsize} ...\end{normalsize}</code>	<code>\normalsize{...}</code>
large	<code>\begin{large} ...\end{large}</code>	<code>\large{...}</code>
Large	<code>\begin{Large} ...\end{Large}</code>	<code>\Large{...}</code>
LARGE	<code>\begin{LARGE} ...\end{LARGE}</code>	<code>\LARGE{...}</code>
huge	<code>\begin{huge} ...\end{huge}</code>	<code>\huge{...}</code>
Huge	<code>\begin{Huge} ...\end{Huge}</code>	<code>\Huge{...}</code>

Colors

By loading the `xcolor` package, you can define your own colors or use the vast selection of default shades. It's as simple as `\textcolor{red}{red text}` to get **red text**. Define your own colors by using `\definecolor{name}{model}{color-spec}`. More information on color definition can be found [here](#).

III Typing math

Let's create a new document to learn how to include math in \LaTeX . Create the necessary header so that it compiles and include this in the preamble `\usepackage{amsmath, amsfonts, amsthm}`. Now you can follow along and compile as we go.

14 Basic math

The most basic way to type math is in-line math. We simply use the dollar sign to enter math mode. Note that either `$ $` or `\(\)` can be used to enter math mode. For example, `$2+2=4$` produces $2 + 2 = 4$. There are a lot of special operators that you can only use in math mode. For example, we can write `$x = \log(\exp(x))$` to get $x = \log(\exp(x))$. Below is a table of very common math commands.

Table 3: Useful operators

Command	Outcome
<code>\exp</code>	\exp
<code>\log</code>	\log
<code>\ln</code>	\ln
<code>\sqrt{x}</code>	\sqrt{x}
<code>\sin</code>	\sin
<code>\tan</code>	\tan
<code>\cos</code>	\cos
<code>\sum</code>	\sum
<code>\prod</code>	\prod
<code>\min</code>	\min
<code>\max</code>	\max
<code>\int</code>	\int
<code>\partial</code>	∂
<code>\lim</code>	\lim
<code>\vee</code>	\vee
<code>\wedge</code>	\wedge
<code>\cup</code>	\cup
<code>\cap</code>	\cap
<code>\to</code>	\rightarrow
<code>\cdot</code>	\cdot
<code>\times</code>	\times
<code>\implies</code>	\implies
<code>\in</code>	\in
<code>\setminus</code>	\setminus
<code>\geq</code>	\geq
<code>\leq</code>	\leq
<code>\neq</code>	\neq
<code>\infty</code>	∞
<code>\emptyset</code>	\emptyset

We can also declare new operators in our preamble by using the function `\DeclareMathOperator{\var}{var}`. This gives us a new command `\var` that produces a plain text version of the variance operator. Compare

$$\text{var}(x) = \sigma^2$$

$$\textit{var}(x) = \sigma^2$$

where the top one is our new operator. We can also produce plain text words in math mode by using `\text{text}`.

Writing all math in-line can make your text cramped and hard to read, so you likely want to mix it up with display math. You can use either `$$ $$` or `\[\]` to enter display math. Switching to display you go from $f(x) = \exp(x)$ to

$$f(x) = \exp(x)$$

In a lot of cases you will want to do this to style, ease of reading, and to emphasize important steps or results.

To add subscripts and superscripts to variables we use the underscore (`_`) and the carrot (`^`). For example, if we want to sum x_i^2 from $i = 1$ to N , we write `\sum_{i=1}^N x_i^2`. Note that we use subscript and superscript on the summation operator to set the bounds on the operation. The inline output is $\sum_{i=1}^N x_i^2$ and the display output is

$$\sum_{i=1}^N x_i^2$$

Notice that with the bounds on the operator, there is a cosmetic difference between the display and the inline presentations. The other thing to note is the use of `{i=1}` to tell the software that the whole segment is the subscript. Omitting the braces is a common mistake with obvious results

$$\sum_i = 1^N x_i^2$$

Fractions are easy to implement. We simply use `\frac{num}{den}` to have the top and bottom parts. Inline this gives us $\frac{x}{y}$ and display

$$\frac{x}{y}$$

While we're looking at fractions, let's consider the following issue.

```
$$ \frac{1}{N} (\sum_{i=1}^N \log(f(\theta|y_i))) + (1-\frac{1}{p})X $$
```

which produces

$$\frac{1}{N} \left(\sum_{i=1}^N \log(f(\theta|y_i)) \right) + \left(1 - \frac{1}{p} \right) X.$$

Look at those terrible parentheses. Unfortunately, \LaTeX doesn't adjust the height of them automatically, so we have to do it ourselves:

```
$$ \frac{1}{N} \left( \sum_{i=1}^N \log(f(\theta|y_i)) \right) + \left( 1 - \frac{1}{p} \right) X $$
```

```
{1}{p}\right)X $$
```

which gives us

$$\frac{1}{N} \left(\sum_{i=1}^N \log(f(\vartheta|y_i)) \right) + \left(1 - \frac{1}{p} \right) X.$$

This also works for brackets and curly braces. However, because braces are special they are written with `\left\{ \right\}`.

Binomials (the choose function) works the same way as fractions. Just replace `frac` with `binom` to get

$$\binom{1}{4}.$$

Greek letters are also important. The command for them simply is a backslash and the name of the letter. This means that ϑ is `\theta` and capitals like Θ is with a capital letter, `\Theta`.

15 Fancy equations

You might want to use numbered equations to draw attention to important results. To get equations with numbers we can use the `equation` environment.

```
\begin{equation}\label{eq:exp}  
F(t) = 1 - \exp(-\lambda t)  
\end{equation}
```

which produces

$$F(t) = 1 - \exp(-\lambda t). \tag{1}$$

Note that the use of the `label` command works exactly like it did with figures and tables so we can refer to Equation 1 by using `\ref{eq:exp}`.

Being able to line up multiple equations creates a nice clean visual effect. The `align` environment provides a system for doing that. The code

```
\begin{align*}  
F(t) &= 1 - \exp(-\lambda t) \\ F(0) &= 1 - \exp(0) \\ &= 0.  
\end{align*}
```

Notice that this combines a few things we have done so far. The `align` removes numbers, just like it did with sections, while the `&` is what lines equations up and the `\\` breaks the line, just like with tables. If we remove the `align` we get equation numbers for each line that can be assigned labels.

Piecewise functions are also important. The `cases` environment inside `math` mode helps us here.

```
$$  
f(t) =
```

```

\begin{cases}
1 - \exp(-\lambda t) & \text{if } t > 0 \\
0 & \text{otherwise.}
\end{cases}

```

The code produces

$$f(t) = \begin{cases} 1 - \exp(-\lambda t) & \text{if } t > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Notice the use of `\text{}` to produce normal-looking text inside math. Also notice that I included the space after “if” because math mode otherwise runs things together without spaces.

16 Matrices

At some point you may find yourself with the inevitable task of writing a matrix. Here’s how you would do that.

```

A =
\begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}

```

And this gives you

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

The `b` in `bmatrix` refers to the brackets. You can also use parentheses. For instance

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$$

is made by

```

A =
\begin{pmatrix}
a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\
a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \ldots & a_{m,n}
\end{pmatrix}

```

```
\end{pmatrix}
$$
```

Besides brackets and parentheses, you can use the `matrix` environment for a plain matrix, `Bmatrix` for a braces, `vmatrix` for vertical lines, and `smallmatrix` for a smaller matrix that can also be used in-line.

17 Theorems and proofs

The theorem and proof environments are the basis for most formal theoretical analysis. You setup these environments by declaring theorems in your preamble.

```
\newtheorem{thm}{Theorem}
\newtheorem{claim}{Claim}
\newtheorem{proposition}{Proposition}
\newtheorem{lemma}{Lemma}
```

The code defines four environments that we can use. Let's go ahead and write a theorem with the following.

```
\begin{thm}\label{th:subsets}
If  $A$  is a subset of  $B$  and  $B$  is a subset of  $C$  then  $A$  is a subset of  $C$ .
\end{thm}
```

Theorem 1. *If A is a subset of B and B is a subset of C then A is a subset of C .*

Now we have the statement of Theorem 1. For the proof we use the proof environment.

```
\begin{proof}
Left as an exercise to the reader.
\end{proof}
```

Proof. Left as an exercise to the reader. □

18 Fancy fonts

Table 18 shows some additional things you may want to use in math mode.

Table 4: Fonts in math mode

Command	Use	Example
<code>\mathbb{}</code>	Blackboard	\mathbb{R} is used to denote real numbers.
<code>\mathbf{}</code>	Boldface	\mathbf{X} bold is sometimes used for matrices.
<code>\mathfrak{}</code>	Fraktur	\mathfrak{R} is also sometimes used to denote real numbers.
<code>\hat{}</code>	Add a hat to a variable	$\hat{\vartheta}$ is an estimate of ϑ .
<code>\tilde{}</code>	Add a tilde	$\tilde{\vartheta}$ is sometimes used to denote a specific value of ϑ .
<code>\bar{}</code>	Add a bar	\bar{x} is often used to denote the mean of x .

IV Presentations with beamer

There are many ways to create slideshows in \LaTeX and customize them. We are going to focus on using the beamer package. It is loaded by specifying beamer as the argument to the `\documentclass{}` command in the your preamble.

Let's start by opening a new file and typing the following

```
\documentclass{beamer}
\author{Rocket J. Squirrel}
\institute{Moose Institute for Toetapping}
\title{Flying Squirrels 'cant Actually Fly}
\date{\today}

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

Not much has changed from making papers. Besides the new document class, we have added the `institute` line. We also have our `maketitle` command wrapped inside something called a `frame`. A `frame` is what beamer calls a slide. When we compile this we get a single slide, which is the title slide.

19 Theme and colors

Right now, your presentation looks a bit boring. You can tweak the look of it by using `\usebeamertheme` to change the layout and `\usecolortheme` to set the color palette. This [website](#) shows various built-in combinations of these two options. For now, we will use the Frankfurt theme and seahorse colors.

```
\documentclass{beamer}
\author{Rocket J. Squirrel}
\institute{Moose Institute for Toetapping}
\title{Flying Squirrels 'cant Actually Fly}
\date{\today}
\usetheme{Frankfurt}
\usecolortheme{seahorse}

\begin{document}
```

```
\frame{\maketitle}  
\end{document}
```

Note that \LaTeX , by default, adds navigation symbols in the bottom right corner of the slide. Since no one ever seems to use those, you should turn them off by adding `\setbeamertemplate{navigation symbols}{}` to your preamble.

20 Frames

We will now add some slides.

```
[...]  
  
\begin{document}  
\frame{\maketitle}  
  
% Navigation choices  
\section{Introduction}  
\stepcounter{subsection}  
  
% slide 1  
\begin{frame}{Introduction}  
This is a great presentation.  
  
We will show many things.  
\end{frame}  
  
% slide 2  
\begin{frame}{Introduction (cont.)}  
  
We will show you  
\begin{itemize}  
    \item Great hypotheses  
    \item Exciting results  
\end{itemize}  
\end{frame}  
  
\end{document}
```

We have now created two slides, both of which are part of the introduction section. The call to `stepcounter` is to tell the software to separate the slides for the navigation bar (the dots at the top in this theme.)

Tables, figures, math, lists, colors, and basically everything else works exactly the same here, so we won't go over them again. Something we have not talked about yet is how to include columns which is probably more popular in presentations than in papers. Columns are just another environment and our new slide will have two columns.

```
[...]
```



```
\end{document}
```

What we did here was open the `columns` environment and then begin and end one column at a time. For each column we also specified how much of the page it will take.

21 Appendix slides

You might want to include additional slides after your main ones. We can do that by using the `\appendix` command. All sections after this declaration are separated from the main slides. This way the additional slides do not show up in the navigation bar. You can stop these extra slides from adding to your page count (in the bottom right corner on most themes) by using the `appendixnumberbeamer` package. If the package is not already part of your \LaTeX distribution, you can download it [here](#).

V Managing citations with `BIBTEX`

Finally, we briefly cover the basics of including citations and references in your compiled document with the aid of `BIBTEX`. You can always manually enter references into your document, but the advantages of using `BIBTEX` are considerable.

1. `BIBTEX` comes bundles with `LATEX`. You don't have to download anything else.
2. In the long run, it will save you a lot of typing.
3. You will not have to type the author, title, date of publication, or page references for the book, article, or unpublished manuscript that you want to cite more than once.
4. `BIBTEX` and `LATEX` will format (and re-format) your references according to whichever bibliography style you require.

22 Manually adding references

We will not go into the details of how `BIBTEX` works, but we will go through the steps that you must follow to use `BIBTEX` to format your citations. Each `BIBTEX` database is a plain text file, and as with `.tex` files, you can create a `.bib` file using any text editor, as long as your references are formatted correctly. For example, within a `.bib` file, a properly formatted reference for a journal article will look like this:

```
@Article{Stokes1963,
  Title = {Spatial Models of Party Competition},
  Author = {Stokes, Donald E.},
  Journal = {The American Political Science Review},
  Year = {1963},
  Month = {June},
  Number = {2},
  Pages = {368--377},
  Volume = {57},
  Publisher = {JSTOR}
}
```

23 Bibliography management

It will save you time to use reference management software like `JabRef` which is [available](#) for Linux, Windows, and Mac. It will format your references for you. If you don't already have it, download it now.¹ Once it is installed, open it and create a new database.

- Click on the green plus sign and choose what kind of entry you are looking for (book, article, PhD thesis...)

¹There are various other bibliography management programs that you can use. Another popular choice is [Mendeley](#) which syncs your bibliography across devices.

- Fill in all of the required fields after clicking on the appropriate tab. If you want to include page references or a volume number, you can enter this information under ‘Optional Fields’.
- Frequently you can download the BibTeX citation for a published book or journal article direction from the publisher’s website. For example, search for Anthony Downs’ *The Economic Theory of Democracy* on Google Scholar. By clicking on ‘Cite’, you will be able to easily locate the BibTeX citation. Copy and paste the citation in the ‘BibTeX Source’ tab on JabRef. Click back to the ‘Required Fields’ tab. Now, all of the fields that can be filled out will have been filled for you. Amazing. Now try going to a journal website and downloading the citation for an article of your choice.
- Once you have filled in all the information, click on the sparkling wand to the bottom left. This will generate a BibTeX key for you which is all the information you will need to cite a reference in your .tex file. The default key pattern is author–year, in our example Downs1957. If you prefer a different one, you can set it yourself in the preferences or manually adjust it.
- To delete a bibliography entry, simply right-click on it and select ‘Delete’.

24 L^AT_EX with BibTeX

Save the bibliography file that you just created in the same folder where you saved the document from Chapter 2. I’m calling my file `texCourse.bib`. To incorporate BibTeX into our document, we will add one line to our preamble and two lines to the end. Those ending lines set the style of our bibliography (in this case the style of the APSR) and then we direct it to our bib file.

```

\documentclass[11]{article}
\usepackage[top=1in,bottom=1in,left=1.25in,right=1.25in]{geometry}
\usepackage{setspace}
\doublespacing

\title{On the Making of Science}
\author{Bullwinkle J. Moose\thanks{Ph.D. Candidate, Moose Institute for
  Toetapping.}}
\date{\today}

\usepackage{natbib} %NEW BIBTEX STUFF

\begin{document}
\maketitle
\begin{abstract}
This paper does stuff
\end{abstract}

\newpage

\section{Introduction}
This paper examines important scientific questions.

```

It does so in a super rigorous and presents results so well-identified that they can't grab a bite without being spotted.

We address all the important concerns with our results and can say with absolute certainty that we have created the last word on this topic.

In this paragraph we want to say just a little more about what a big deal the paper is.

This paper advances all of science by about 25 years.

In fact, we have produced 200\% more science than anyone else writing on this topic.

```
\section{Results}
\begin{table}[h]
  \centering
  \caption{Table of stuff}
  \label{tab:Stuff}
  \begin{tabular}{rc|l}
    \hline
    Input 1 & Input 2 & & Sum \\ \hline
    5 & 10 & & 15 \\
    2 & 9 & & 11 \\ \hline
  \end{tabular}
\end{table}
```

Table [\ref{tab:Stuff}](#) shows just all the cool science we do. [fig:plots](#) graphs some other stuff.

```
\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{3dPlots.pdf}
\caption{Some plots}
\label{fig:plots}
\end{figure}
```

That figure sure was a pretty. [\footnote{The prettiness of the figure was rated as 7.9 on the princess beauty scale.}](#)

```
% BIBTEX STUFF
\bibliographystyle{apsr}
\bibliography{texCourse}

\end{document}
```

Citing commands

We're now ready to do some citing. Table 5 shows a bunch of different ways to cite.

Table 5: Cite commands

Command	Output
<code>\citet{Downs1957}</code>	Downs (1957)
<code>\citet{Downs1957, Stokes1963}</code>	Downs (1957); Stokes (1963)
<code>\citep{Downs1957}</code>	(Downs, 1957)
<code>\citeauthor{Downs1957}</code>	Downs
<code>\citeyear{Downs1957}</code>	1957
<code>\citep[p. ~215]{Downs1957}</code>	(Downs, 1957, p. 215)
<code>\citep[see] [215]{Downs1957}</code>	(see Downs 1957, 215)

The use of brackets adds notes to the citations. One set of brackets adds text to the end of the cite while two sets of brackets sandwiches the citation between the two bits of text. Other cite commands can be found [here](#) if you need more flexibility.