



Introduction to \LaTeX

Mr. Rusty Laracuenti

Updated September 29, 2015

Abstract

This is an initial attempt to create a compact introduction to Overleaf and \LaTeX in general. Topics and examples are tailored to meet the needs of The Hudson School. \LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. \LaTeX is the de facto standard for the communication and publication of scientific documents.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 3 |
| 2 | The Basics | 3 |
| 2.1 | Special Characters | 4 |
| 2.2 | Bold, Italic, and Underline | 4 |
| 2.3 | Sections, Subsections | 5 |
| 2.4 | Spacing | 5 |
| 3 | The Essentials | 6 |
| 3.1 | Figures and Images | 6 |
| 3.2 | Lists | 8 |
| 3.3 | Tables | 9 |
| 3.4 | Comments | 10 |
| 3.5 | Colors | 10 |
| 3.6 | Boxes and Frames | 11 |
| 4 | The Good Stuff | 14 |
| 4.1 | Spacing and Text | 14 |
| 4.2 | Subscripts, Superscripts and Fractions | 16 |
| 4.3 | Numbering and Alignment | 17 |
| 4.4 | Special Symbols and Greek Letters | 19 |
| 5 | The Bibliography | 19 |
| 5.1 | Setup | 20 |
| 5.2 | Building the Database | 20 |
| 6 | The Advanced Stuff | 21 |
| 6.1 | Custom Commands | 21 |
| 6.2 | Tikz Package | 22 |
| 6.3 | Octave and Matlab | 22 |
| A | Examples by Subject | 28 |
| A.1 | Calculus | 28 |
| A.2 | Physics | 28 |
| A.3 | Chemistry | 29 |
| A.4 | Biology | 30 |

List of Figures

| | | |
|---|---|----|
| 1 | A funny physics cartoon | 8 |
| 2 | A custom blank graph with GeoGebra. | 23 |
| 3 | L ^A T _E X code generated with Octave. | 24 |
| 4 | Diagram with circuitikz and tikz package. | 26 |
| 5 | Unit Circle with Tikz | 27 |
| 6 | Branched Molecules | 30 |

1 Introduction

This document was created using the standard *article* class suitable for most academic articles and reports. In Overleaf there are two main windows. The window on the left displays your raw \LaTeX source code. The window on the right is the preview window. If the preview window is set to auto then the window will be updated in real time, however, there is a slight delay. The preview window shows you what your finished document will look like after you have exported it to a PDF.

In Overleaf, clicking on an object or section in the preview window will automatically move your cursor to the corresponding place in the source code window. The source code window can be viewed as raw source code or as “Rich Text.” Rich text mode shows a more familiar version of the document similar to typing into a standard word processor.

Everything in \LaTeX is done with commands. Commands always start with a backslash and are case sensitive. Certain commands can only be used with the inclusion of certain packages or within certain environments. The point of \LaTeX is to take the focus off of the formatting of documents so the user can concentrate on the content. This being said, once you have settled on a format or template, making small stylistic changes can be difficult.

This document was created in \LaTeX for a reason. It is much easier to start with a working example and make adjustments than to create from scratch. When viewing this document on Overleaf you can view the source code and output simultaneously. This means any special formatting, tables, and math that you see can be copied and imported into your own document. Importing these examples into your \LaTeX document is as simple as copying the source code and pasting it into your document. You can then change the values and adjust as needed.

When viewing this document as a PDF most of the example source code is included verbatim in the peach colored boxes. As a PDF, you can also click on hyperlinks to be taken directly to that page on the internet. Clicking on an item in the table of contents will also bring you to that specific section.

2 The Basics

This is by no means meant to be a comprehensive introduction to \LaTeX . There are many document classes and thousands of packages. Extensive documentation is also available freely online. You can only use one document class per document but you can import as many packages as you need. Packages add extra features and capabilities to your document.*

The bare minimum code needed to create a \LaTeX document is as follows:

*Some packages may not play well with others and some have to be loaded in a particular order. The packages loaded in the preamble of your lab template should be sufficient for most cases.

```
\documentclass{...}

...your preamble goes here...

\begin{document}

...your text goes here...

\end{document}
```

The entire text of the final product is located between the `\begin{document}` and the `\end{document}` command. The space between `\documentclass{}` and `\begin{document}` is called the preamble. The preamble is where you tell \LaTeX what kind of document to create. Some well known document classes include article, book, exam, letter, and beamer. The document class governs the overall format of the document such as fonts and default margins.

The preamble is also where you import your packages. Certain packages are mandatory to enable certain features such as the ability to add images to your document. You can take a look at the preamble of this document to see what packages are included and for what reason.*

2.1 Special Characters

Certain characters in \LaTeX have special meaning. \LaTeX interprets these symbols as an action to be executed. When \LaTeX does not recognize your symbol as a legitimate command you will get an error and the document will not compile. Any of the following characters will cause this to happen.

```
# $ % & ~ _ ^ \ { }
```

There is, of course, a way around this. Placing a backslash before most of the characters will cause them to be displayed as ordinary symbols. The need to talk about `$$` or `%` understandably comes up often. This works for all of the symbols except for the backslash since the double backslash is, in itself, a command. To get a `\` to appear in your document type `$$\backslash$`. In most cases if you really just need a slash in your document a forward slash (`/`) will usually suffice.

2.2 Bold, Italic, and Underline

Bold, *italic*, and underline each have a command. Overleaf makes it even easier by enabling keyboard shortcuts for boldfaced and italic type. Highlighting a word and pressing `Ctrl` + `B` or `Ctrl` + `I` causes the word to be made bold and italicized, respectively. The raw commands are

```
\textbf{bold text}
\textit{italic text}
\underline{underlined text}
```

*Some of these packages are only necessary for this document and may not be included in your lab template.

2.3 Sections, Subsections

L^AT_EX automatically handles section numbering and formatting. This allows the user to concentrate on content rather than formatting. As new sections are added L^AT_EX will adjust section numbering accordingly. The code for creating sections and subsections is as follows:

```
\section{Name of Section}
\subsection{Name of Subsection}
```

It is also possible to create sections without numbering them. This is achieved by putting an asterisk after the section command.

```
\section*{Name of Section Without Numbering}
```

Subsection without Numbering

The next numbered section will be numbered as if this section doesn't exist. Sections without numbers, by default, will not appear in the table of contents.

2.4 Spacing

L^AT_EX also handles spacing of words, paragraphs and equations automatically. This can be confusing, at first, to someone familiar with a word processor. This is better illustrated with an example:

```
This
is
the
beginning of a
\LaTeX{}
tutorial.
```

Produces ...

This is the beginning of a L^AT_EX tutorial.

Now, suppose for some odd reason you did in fact want the text to show as written. To force L^AT_EX to end a line you use a double backslash. To achieve the text as it looks you need to manually end each line.

```
This \\
is \\
the \\
beginning of a \\
\LaTeX{} \\
tutorial.
```

Produces ...

This
is
the


```
\includegraphics[width=8cm]{cartoon}
```



"I'm warning you, Perkins - your flagrant disregard for the laws of physics will not be tolerated!"

In addition to width one can specify length, scale, angle of rotation, and the resolution of the image. \LaTeX understands a number of different units including, but not limited to,

mm Millimeters

cm Centimeters

in Inches

pt A pt is approximately a third of a millimeter

We have included an image but in order to number it we must tell \LaTeX that it is indeed a figure. Environments are used to change the formatting of a block of text without affecting the format of the overall document. In order to turn our cartoon into a figure we use the figure environment to give it a number handled by \LaTeX . To center our figure we use the center environment.

Figures can be referred to later by cross-referencing. The *label* command allows you to reference the figure later even if the number and page have changed. Such as refer to Figure 1 on page 8.

```
\begin{figure}[!ht]
\begin{center}
\includegraphics[width=5cm]{cartoon}
\caption{A funny physics cartoon}
\label{fig:toon}
\end{center}
\end{figure}
```



Figure 1: A funny physics cartoon

With the `ref` and `pageref` command you can refer to the figure later with `\ref{fig:toon}` and `\pageref{fig:toon}`

3.2 Lists

Environments are available for making different kinds of lists including bulleted lists and numbered lists. The `enumerate` environment is used to create a numbered list.

```
\begin{enumerate}
\item Item 1
\item Item 2
\end{enumerate}
```

1. Item 1
2. Item 2

A bulleted list can be created with the `itemize` environment.

```
\begin{itemize}
\item Item 1
\item Item 2
\end{itemize}
```

- Item 1
- Item 2.

A list of definitions can be achieved with the `description` environment.

```
\begin{description}
\item[Word] Definition
\item[Concept] Explanation
\item[Idea] Longer explanation
\end{description}
```


Word Definition

Concept Explanation

Idea This is what will happen if your definitions are rather lengthy and take up multiple lines.

3.3 Tables

Tables are made using the *tabular* environment. Tables can also be numbered and cross-referenced as figures such as Table 1 on page 9.

| Circuit | Voltage | Resistance | Current |
|-----------|---------|------------|---------|
| Circuit 1 | 10 V | 10 Ohms | 1 Amps |
| Circuit 2 | 50 V | 5 Ohms | 10 Amps |
| Circuit 3 | 20 V | 2 Ohms | 20 Amps |
| Circuit 4 | 9 V | 3 Ohms | 3 Amps |

Table 1: Table created using the *tabular* environment.

```
\begin{table}[h!]  
\begin{center}  
  \begin{tabular}{| l | | c | c | c |}  
  \hline  
  Circuit & Voltage & Resistance & Current \\ \hline  
  Circuit 1 & 10 V & 10 Ohms & 1 Amps \\ \hline  
  Circuit 2 & 50 V & 5 Ohms & 10 Amps \\ \hline  
  Circuit 3 & 20 V & 2 Ohms & 20 Amps \\ \hline  
  Circuit 4 & 9 V & 3 Ohms & 3 Amps \\ \hline  
  \end{tabular}  
\end{center}  
\caption{Table created using the tabular environment.}  
\label{tab:circuitgrid}  
\end{table}
```

| Circuit Number | Voltage | Resistance | Current |
|----------------|---------|------------|---------|
| Circuit 1 | 10 V | 10 Ohms | 1 Amps |
| Circuit 2 | 50 V | 5 Ohms | 10 Amps |
| Circuit 3 | 20 V | 2 Ohms | 20 Amps |
| Circuit 4 | 9 V | 3 Ohms | 3 Amps |

Table 2: A table with fewer gridlines.

Creating tables in \LaTeX can be daunting as you can tell by the code for Table 1. There are, however, online table generators that will create the \LaTeX table code for you and are more user friendly. A quick google search for “ \LaTeX table generator” yielded

<http://www.tablesgenerator.com/>
and
<http://truben.no/table/old/>

You can always create a table in any way you like and import as a picture with the *includegraphics* command.

3.4 Comments

Comments, generally, are used to leave notes for yourself within the source code. They do not appear in the official document. Most programming languages have a way of excluding certain text from being executed as normal code.

As projects get large, commenting becomes increasingly necessary. A percentage sign is used to designate comments or code which should not be executed.

```
% Find evidence to support this section
```

Comments are also useful for organizing sections of a larger project to make the code easier to read.

```
%-----  
% Section: Introduction  
%-----
```

Another good use for comments is to have a section of common commands and unusual or hard to remember syntax as comments in a default blank document file that you keep handy. This way you can keep certain commands handy in every new document you create, eliminating the need to search or refer to this document each time. It is always a good idea to leave yourself some notes especially when you have to walk away from a project for the night.

There is a special package which enables you to leave comments that are visible on the published document. You are able to leave comments inline with your text or in the margin of the document. Comments can be added to the margins of the document using the *todo* command, as shown in the example on the right. You can also add inline comments and specify a color.

```
...or for teachers to leave comments on student drafts.
```

```
\todo{As the command implies this may be useful for  
keeping track of things left unfinished.}
```

```
\todo[inline, color=green!40]{\dots or for  
teachers to leave comments on student drafts.}
```

```
Hmm ... this may be a good time to talk a bit about color ...
```

As the command implies this may be useful for keeping track of things left unfinished.

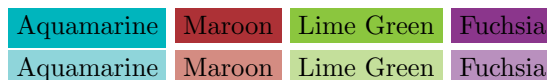
3.5 Colors

If you don't specify a color the comment defaults to the orange seen in the margin comment. Default colors that can be called by name without the need for additional packages are white, black, red, green, blue, cyan, magenta, and yellow.

The package *xcolor* will allow you to specify an additional sixty eight colors by name. Please see the appendix for a complete list of colors available in the *xcolor* package. Colors such as **Aquamarine**, **Maroon**, **Lime Green**, and **Fuchsia** (assuming you can remember how to spell it) become available. This is achieved by placing the following line in the preamble.

```
\usepackage[usenames,dvipsnames]{xcolor}
```

Colors can be used to highlight text with the use of the *colorbox* command. The opacity of colors can be changed by adding an exclamation point and a number afterwards. The second set of color boxes are set to fifty percent opacity.



```
\begin{center}
\colorbox{Aquamarine}{Aquamarine}
\colorbox{Maroon}{Maroon}
\colorbox{LimeGreen}{Lime Green}
\colorbox{Fuchsia}{Fuchsia}\
\colorbox{Aquamarine!50}{Aquamarine}
\colorbox{Maroon!50}{Maroon}
\colorbox{LimeGreen!50}{Lime Green}
\colorbox{Fuchsia!50}{Fuchsia}\
\end{center}
```

Custom Colors

Colors can also be defined manually by their RGB value. That is their percentage of red, green, and blue. All colors can be defined in this way. Black is the absence of any color and white is the inclusion of all colors. You can define your own custom color names to use later with their RGB values as well.

```
\definecolor{pukegreen}{rgb}{0.53, 0.66, 0.42}
\definecolor{poopbrown}{rgb}{0.59, 0.44, 0.09}
```

Now **pukegreen** and **poopbrown** can be called by name when needed.*

3.6 Boxes and Frames

There are many methods for creating boxed text and equations in L^AT_EX. One of the simplest ways is the *framebox* command.

```
\begin{center}
\framebox{$e=mc^2$}
\end{center}
```

$$e = mc^2$$

*An extensive list of colors and their RGB values is available at <http://latexcolor.com>

The `framebox` command can be used `in line with text` or in the center environment as demonstrated above. Here it is used in the equation environment so it can be numbered and referenced later.

```
\begin{equation}
\framebox{$\vec{F}=m\vec{a}$}
\label{eq:newton2nd}
\end{equation}
```

$$\vec{F} = m\vec{a} \tag{1}$$

You can specify the width of a `frame box` by adding a width parameter in square brackets. To create a colored box use the `fcolorbox` command.

```
\framebox[3cm]{framebox}
\fcolorbox{Fuchsia}{white}{fcolorbox}
```

Where fuchsia is the color of the frame and white is the color of the background. It is possible to change both the thickness of the box and distance between the box and its contents. Issuing the following commands will change the look of all frame boxes from that point on.

```
\setlength{\fboxsep}{3mm}
\setlength{\fboxrule}{1mm}
```

All `frame boxes` now appear like this by default. The setting also affects

`color boxes`. All of our boxes have contained only one line of text although it is possible to create an itemized or numbered list within a box. Notice the `itemize` environment has changed to `Bitemize`.

```
\begin{center}
\fbbox{Dr. Seuss Titles: \hspace{1cm}}
\begin{Bitemize}[t]
\item Green Eggs and Ham
\item The Lorax
\item The Cat in the Hat
\end{Bitemize}}
\end{center}
```

| | |
|-------------------|----------------------|
| Dr. Seuss Titles: | • Green Eggs and Ham |
| | • The Lorax |
| | • The Cat in the Hat |

Once you realize that the rest of your document after this point has been affected you may want to set these values back to their default settings which appears to be

```
\setlength{\fboxsep}{3pt}
\setlength{\fboxrule}{0.5pt}
```

This document also uses two additional packages for creating boxes and frames: *fancybox*^{*} and *mdframed*[†]. The *fancybox* package allows four additional types of boxes. These are shadow box, double box, oval box and a thicker version of an oval box.

```
\begin{center}
\shadowbox{This}
\doublebox{that}
\ovalbox{and}
\Ovalbox{the other thing.}
\end{center}
```

This, that and the other thing.

Package: mdframed

The other package, called *mdframed*, is useful for aesthetically pleasing boxes that hold a lot of text.

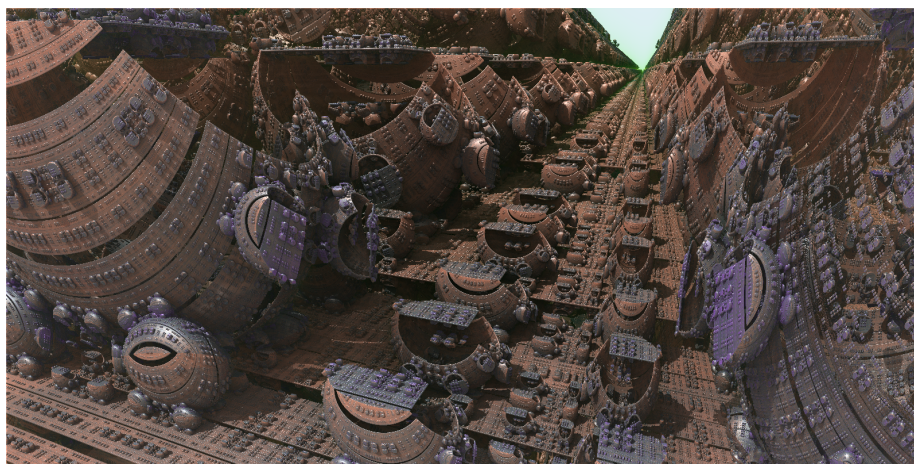
The *mdframed* package is extensive and adds a lot of options for creating boxes. In this document it was used to set up a default box for the verbatim environment. The verbatim environment is how the raw latex code is put into the document without it being executed as commands. The properties of the box need to be edited only once in the preamble to change their appearance throughout the document. This could be incredibly useful if you wanted to have all of your equations wrapped in the same kind of box.

Previous box created with the following options:

```
\begin{mdframed}[backgroundcolor=Magenta!40,shadow=true,
roundcorner=5pt,frametitle={Package: mdframed}]
...the text...
\end{mdframed}
```

^{*}For more information, see the *fancybox* documentation at <http://mirrors.ctan.org/macros/latex/contrib/fancybox/fancybox-doc.pdf>

[†]For more information, see the *mdframed* documentation at <http://ctan.math.washington.edu/tex-archive/macros/latex/contrib/mdframed/mdframed.pdf>



4 The Good Stuff

Have you ever wondered how text books and some websites get their equations to show up perfectly? There are numerous environments for typesetting math in \LaTeX . The most common are the math environment, the displaymath environment and the equation environment. The math environment allows you create equations and functions inline with your text. The displaymath environment will center an equation on its own line. The equation environment is used to display important equations you may wish to reference later.

There are short hand methods to enable the math and displaymath environments. Inline math and the displaymath environments are activated by $\$$ and $\$\$$ respectively. Recall that environments must have a beginning and an end command so you must wrap the single dollar signs around your inline equation and the double dollar signs around your displaymath equation.

The equation environment is used to create numbered equations similar to figures that can be referenced later. There is no shorthand command to enable this environment and must be invoked with the usual $\backslash\text{begin}\{\text{environment}\}$ and $\backslash\text{end}\{\text{environment}\}$ commands. In this case, of course, you would use $\backslash\text{begin}\{\text{equation}\}$ to enable the equation environment.

4.1 Spacing and Text

\LaTeX usually handles spacing in math mode without any issues but certain symbols may need a little help adjusting to our aesthetic convention. Refer to table 3 for a list of spacing commands in math mode.

Manual spacing is needed in certain quirky situations especially when dealing with text in math mode. When using inline math mode it is usually sufficient to exit in and out of math mode for text but in display math mode this wouldn't be possible. Text just does not display *properly in math mode*. Suppose you want to list two solutions as,

$$x = 4 \quad \text{and} \quad x = -2$$

*<http://texblog.org/2014/04/09/whitespace-in-math-mode/>

| Name | Command | Example |
|------------------|---------------------------|----------------------------------|
| default space | | $abc \rightarrow \leftarrow abc$ |
| thin space | <code>\,</code> | $abc \rightarrow \leftarrow abc$ |
| thin neg. space | <code>\!</code> | $abc \rightarrow \leftarrow abc$ |
| medium space | <code>\:</code> | $abc \rightarrow \leftarrow abc$ |
| large space | <code>\;</code> | $abc \rightarrow \leftarrow abc$ |
| 0.5em space | <code>\enspace</code> | $abc \rightarrow \leftarrow abc$ |
| 1em space | <code>\quad</code> | $abc \rightarrow \leftarrow abc$ |
| 2em space | <code>\qquad</code> | $abc \rightarrow \leftarrow abc$ |
| custom space | <code>\hspace{3em}</code> | $abc \rightarrow \leftarrow abc$ |
| fill empty space | <code>\hfill</code> | $abc \rightarrow \dots$ |

Created by <http://texblog.org>

Table 3: Spacing in Math Mode.*

\$\$

$x=4$ and $x=-2$

\$\$

produces ...

$$x = 4 \text{ and } x = -2$$

To achieve what you want use the `text` command to make the text appear normal font and manual spacing to give it some room.

\$\$

$x=4 \quad \text{\texttt{\code{\text\{and\}}}} \quad x=-2$

\$\$

Or to assume n is an odd integer,

$$\text{then } n = 2k + 1, \text{ where } k \in \mathbb{Z}$$

use...

```
$$  
\text{Then} \; ; n=2k+1, \;, \text{where} \;, k \in \mathbb{Z}  
$$
```

Similar to text mode, putting equation elements on separate lines is not going to override the automatic spacing such as equation 2.

```
\begin{equation}  
e  
^{\;i\pi}  
=  
-1  
\end{equation}
```

Still produces ...

$$e^{i\pi} = -1 \tag{2}$$

4.2 Subscripts, Superscripts and Fractions

Typesetting math in L^AT_EX is similar to entering an equation into an online solver like Wolfram Alpha. Subscripts are created with an underscore and superscripts are created with a “carrot.” The formula for the slope of a line is $m = (y_2 - y_1)/(x_2 - x_1)$.^{*} The code is as follows:

```
$ {m=(y_2-y_1)/(x_2-x_1)} $
```

The same equation using the displaymath mode looks like:

$$m = (y_2 - y_1)/(x_2 - x_1)$$

The only difference in the code here is \$\$ instead of \$. Typesetting the equation in this way requires the use of the parentheses to prevent the math gods from turning over in their graves. The difference between $x/2$ and $\frac{x}{2}$ is the use of *frac* command. To format a fraction with a horizontal division symbol use the following syntax within your chosen math environment.

```
\frac{Numerator}{Denominator}
```

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Fractions can, of course, be nested as well...

```
x=\frac{1+\frac{1}{n}}{n!}
```

^{*}To keep an inline math equation from breaking across multiple lines place an extra set of { } around the equation.

produces, $x = \frac{1 + \frac{1}{n}}{n!}$

The presence of the curly brackets should not be overlooked as they are used frequently in all areas of L^AT_EX. They are used like standard parentheses in all situations where you want a command to apply to multiple characters. There is a major difference between e^2x and e^{2x} but the difference in the code is subtle and a matter of remembering to use curly brackets.

```
$e^2x$ vs. $e^{2x}$
```

Keep in mind that fractions and certain symbols will either appear distorted or create additional space in your paragraph when using inline math mode. By default L^AT_EX will prioritize line spacing and adjust the equation accordingly. $m = \frac{y_2 - y_1}{x_2 - x_1}$ is the default formatting for inline equations. The best practice is to keep inline math mode reserved for special characters and simple one line equations. The command *displaystyle* can be used to change this default behavior. $m = \frac{y_2 - y_1}{x_2 - x_1}$ is how inline math equations will look once the *displaystyle* command has been used within math mode. Notice the line spacing has been adjusted to allow the equation to display properly. To enable this for a single equation use the command in math mode,

```
$ \displaystyle m = \frac{y_2 - y_1}{x_2 - x_1} $
```

To enable this for every inline equation by default place the following line in the preamble of the document.

```
\everymath{\displaystyle}
```

4.3 Numbering and Alignment

To refer to our slope equation later we can number and label it using the equation environment in the same way as tables and figures.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{3}$$

Equation 3 is the equation for the slope of a line between two points (x_1, y_1) and (x_2, y_2) .

Note that the equation environment is for assigning a single number to an equation or a derivation.* The code for equation 3 is as follows:

```
\begin{equation}
m = \frac{y_2 - y_1}{x_2 - x_1}
\label{eq:slope}
\end{equation}
```

Care should be taken when assigning labels to equations, figures, and tables. The purpose of the *label* and *ref* commands is that L^AT_EX automatically takes care of the numbering as changes are made. Therefore, it is foolish to label

*Using the split environment within the equation environment will allow for multiple lines.

our tables as table1 or table2. A better method is to be descriptive. Such as the label given to equation 3: “eq:slope.” As another example, the keyword “eq:einstein” is an easy way to refer to equation 4.

$$e = mc^2 \tag{4}$$

For equations on multiple lines or derivations use the align environment with the use of & to define where they are aligned vertically. The align environment, by default, will number each equation. To avoid this, similar to section numbering, use an asterisk after align in the declaration. To align equations vertically with respect to the equal sign:

```
\begin{align*}
x^2-x-20 &= 0 \\
(x-5)(x+4) &= 0 \\
x &= 5, -4
\end{align*}
```

$$\begin{aligned} x^2 - x - 20 &= 0 \\ (x - 5)(x + 4) &= 0 \\ x &= 5, -4 \end{aligned}$$

If no & is used the environment will default to aligning the equations on the right. The align environment will give a unique number to each equation. The previous result using the align environment with numbering looks like:

$$\begin{aligned} x^2 - x - 20 &= 0 && (5) \\ (x - 5)(x + 4) &= 0 && (6) \\ x &= 5, -4 && (7) \end{aligned}$$

This is not a good use of the align environment since these are not equations that are distinct and need to be referred to separately. In most situations you may want only the result to be numbered. If only one of these lines is to be numbered then the command *nonumber* should be used before the end line command. For example:

```
\begin{align}
x^2-x-20 &= 0 \nonumber \\
(x-5)(x+4) &= 0 \nonumber \\
x &= 5, -4 \label{eq:quad_solution}
\end{align}
```

produces ...

$$\begin{aligned} x^2 - x - 20 &= 0 \\ (x - 5)(x + 4) &= 0 \\ x &= 5, -4 \end{aligned} \tag{8}$$

There is a subtle but important distinction to be made between the numbering styles of the equation and the align environments. The equation environment is used to create a single numbered equation even if this equation spans multiple lines. Using the split environment within the equation environment will allow for multiple lines aligned vertically by & as in the align environment. In the previous example referring to Equation 8 refers specifically to the solution and not the problem. You can give the group of equations a number using the equation environment as follows:

```
\begin{equation}
\begin{split}
x^2-x-20 &= 0 \\
(x-5)(x+4) &= 0 \\
x &= 5, -4
\end{split}
\label{eq:quad_problem}
\end{equation}
```

$$\begin{aligned}
 x^2 - x - 20 &= 0 \\
 (x - 5)(x + 4) &= 0 \\
 x &= 5, -4
 \end{aligned}
 \tag{9}$$

A more useful example of this type of numbering is to number an equation or result that is shown as a derivation.

4.4 Special Symbols and Greek Letters

Greek letters can be used in math mode simply by typing their names preceded by a backslash. Lowercase greek letters like α , β , γ , θ , and ϵ are typed in **math mode** as follows:

```
$ \alpha, \beta, \gamma, \theta, and \text{and} \epsilon $
```

Uppercase greek letters that differ from our normal uppercase letters are produced by capitalizing the first letter in the name of the symbol. For instance, alpha, beta and epsilon are simply A, B, and E respectively. Uppercase gamma and theta are unique and Γ and Θ are produced as follows:

```
$ \Gamma $ and $ \Theta $
```

5 The Bibliography

Although L^AT_EX has its own embedded environment for writing bibliographies it is beneficial in the long run to use BibTeX.* BibTeX is a tool and a system for generating a list of references from a database of sources. The database lives in a text file which can be centralized and therefore maintained by a person, a class, or an entire department. More than one file can be used so the sources can be compartmentalized or organized by subject. This means a source only has to be typed once.

*For more information on BibTeX visit http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management

5.1 Setup

The database is formatted and housed in a separate `*.bib` file. The bibliography will automatically generated and formatted using only the sources cited in the document.

Each source in the database must be given a unique reference number in order to be cited. Remembering a reference number for each source may not be feasible if not planned out in advance. With a little foresight you can systematize the naming convention to make things easier. An example of such a system is to use the first three letters of the authors last name and the last two digits of the year of publication for the reference number.

Once cited, the source will automatically be included in the bibliography. To cite the source of a quote or a specific paragraph you use the `cite` command and the reference number as follows. There are numerous options for formatting the type of citation. Once a format has been set to default there is no need to manually format each citation.

```
\cite{roy01}
```

This is used when you want the citation to appear visible. For more information on finite element analysis refer to [8]. To include a general source in your bibliography without attributing it to a particular passage, idea or equation you use the `nocite` command in the same manner. It is probably good practice to keep all your `nocited` sources in a logical place. One such place could be towards the end of the document before the bibliography in a section separated by comments.

```
%-----  
% Sources to be included even if not cited  
%-----  
\nocite{source1}  
\nocite{source2}  
\nocite{source3}  
%-----
```

An example entry in the database for citing a book:

```
@book{strang2007computational,  
  title={Computational science and engineering},  
  author={Strang, Gilbert},  
  volume={1},  
  year={2007},  
  publisher={Wellesley-Cambridge Press Wellesley}  
}
```

5.2 Building the Database

Google Scholar has a way to export citations in BibTeX form. This must be enabled in the settings for Google Scholar before the option will appear. If you are signed into the google system there should be no need to sign into Google Scholar separately.

Search for a source and simply click "Import into BibTeX" underneath the article summary on the search page. Copy that text into your class database and adjust the reference number to conform to the agreed upon system.

There are many online resources that can export citations in BibTeX form and even create a *.bib file for you upon request. It is left to the reader to discover what works best for them.

6 The Advanced Stuff

6.1 Custom Commands

New commands are easily created in L^AT_EX. New commands, of course, cannot conflict with existing L^AT_EX commands. Custom commands are useful for a variety of reasons. Commands can be declared anywhere in the document but for obvious reasons you should keep these tucked away someplace safe like the preamble of your default document. The command must be declared before it can be used. The syntax is simple:

```
\newcommand{New_Command}{Old_Command}
```

A skillful administrator might create new commands for commonly used phrases or even paragraphs in official correspondences.

```
\newcommand{\accepted}{Our usual spiel.}
\newcommand{\yadayada}{Our closing spiel.}
```

Writing a typical letter would then be as simple as

```
Dear So and So, \\
\accepted \\
\yadayada \\
From, \\
VIP
```

Commands can also accept arguments such as

```
\newcommand{\lamb}[1]{#1 had a little lamb}
\newcommand{\hadlittle}[2]{#1 had a little #2}
```

and typing

```
\lamb{Mary} and \hadlittle{John}{dog}.
```

Produces, Mary had a little lamb and John had a little dog.

The need to type `\begin{environment}` and `\end{environment}` can get tedious if you use a certain environment often enough. It is possible to define a custom command that will wrap your input in a particular environment. An editor with auto-complete is the preferred method but in a pinch:

```
\newcommand{\bc}[1]{\begin{center} #1 \end{center}}
```

Allows you to use

```
\bc {some centered text.}
```

Instead of

```
\begin{center}  
some centered text.  
\end{center}
```

to achieve

some centered text.

6.2 Tikz Package

The Tikz package is useful for drawing diagrams, graphs, and so many other things. Learning the ins and outs of the tikz package, however, is a daunting task. The user manual is over a thousand pages, but the capabilities are quite amazing.

The good news is that there are user friendly programs for creating diagrams that can export the tikz code for you. One of these is the open source program, GeoGebra.*

With Geogebra, creating plots and graphs is simple and its export to tikz dialog is also easy to use. The code is not easy on the eyes but the diagram is great.

One thing that is fairly common is the need to insert blank coordinate axes. So why not make it a new command ...

```
\newcommand{\axes}{  
Tikz code from Geogebra  
}
```

Now inserting a blank graph into a document is as easy as typing `\axes` to make Figure 2. With a bit of skill you could add arguments to the command to adjust the range and domain.

The *circuitikz* package with the *siunitx* options adds the ability to create circuit diagrams with relative ease as in Figure 4. This is done by adding the following line to the preamble.

```
\usepackage[siunitx]{circuitikz}
```

6.3 Octave and Matlab

Matrices can be tedious to type in L^AT_EX to say the least. The code for the matrix in Figure 3 is as follows.

*Visit <https://www.geogebra.org/> for more information.

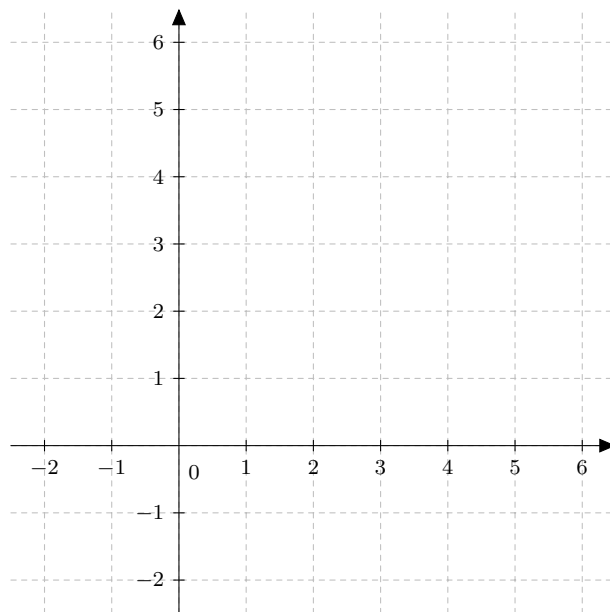


Figure 2: A custom blank graph with GeoGebra.

```


$$\begin{bmatrix}
2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2
\end{bmatrix}$$


```

$$\begin{bmatrix}
2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2
\end{bmatrix}$$

Figure 3: L^AT_EX code generated with Octave.

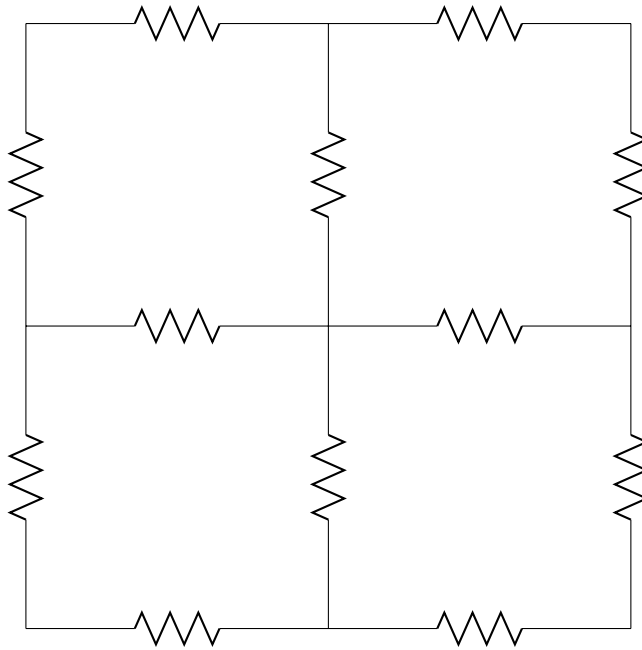


Figure 4: Diagram with circuitikz and tikz package.

References

- [1] <http://bay.uchicago.edu/tex-archive/macros/latex/contrib/fancybox/fancybox-doc.pdf>, feb 2015. fancybox.
- [2] Latex color definitions. <http://latexcolor.com/>, jan 2015.
- [3] Latex documentation. <http://latex-project.org/guides/>, jan 2015. Good site for tutorials.
- [4] Latex/boxes - wikibooks, open books for an open world. <http://en.wikibooks.org/wiki/LaTeX/Boxes>, feb 2015. box wiki.
- [5] <https://www.tug.org/TUGboat/tb28-1/tb88f1om.pdf>, feb 2015.
- [6] Ron Larson and Bruce Edwards. *Calculus*. Cengage Learning, 2009.
- [7] MK MK. Create your own beamer template. <http://programming-r-pro-bro.blogspot.com/2011/11/create-your-own-beamer-template.html>, jan 2015.
- [8] David Roylance. Finite element analysis. *Department of Materials Science and Engineering Massachusetts Institute of Technology, Cambridge, MA*, 2139, 2001.
- [9] Gilbert Strang. *Computational science and engineering*. Wellesley-Cambridge Press Wellesley, 2007.

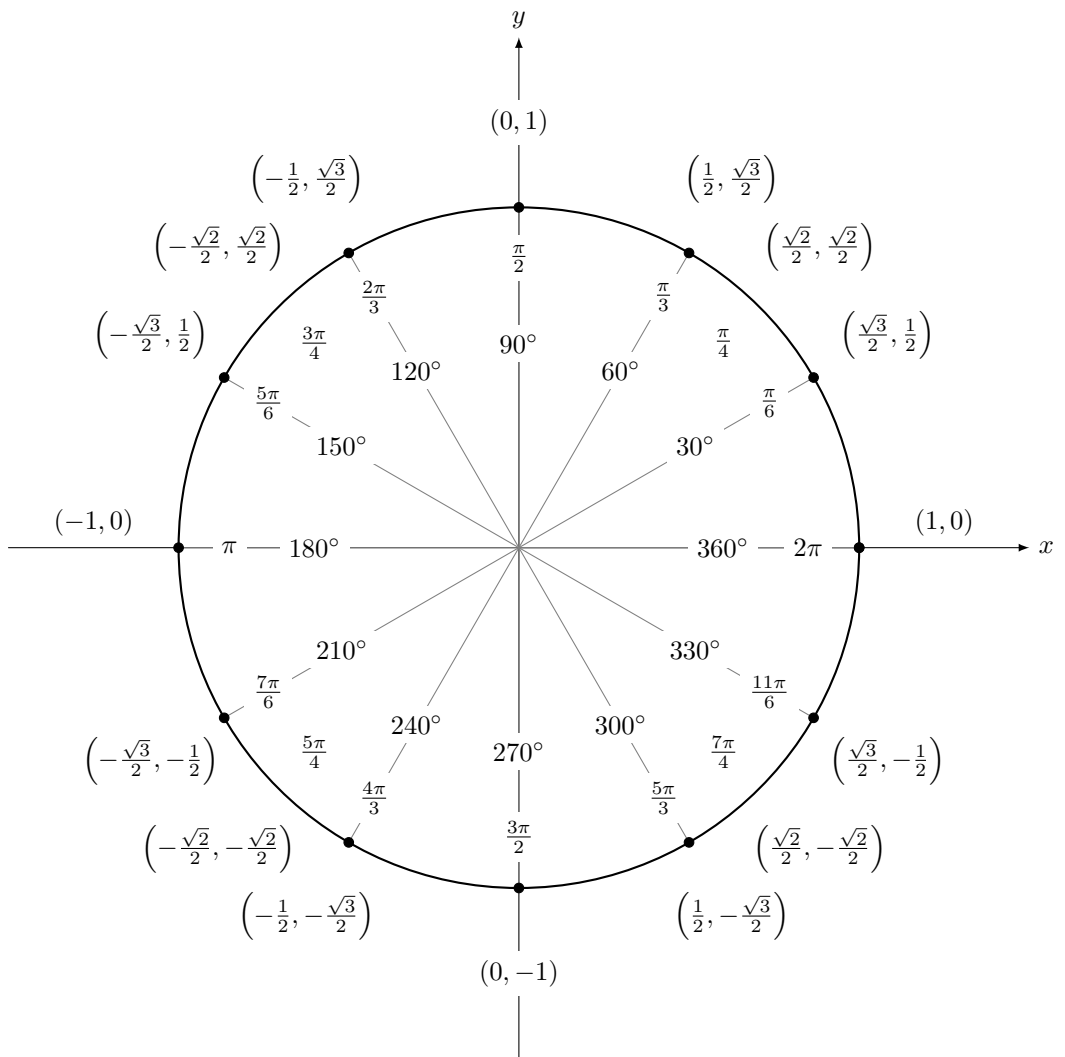


Figure 5: Unit Circle with Tikz

A Examples by Subject

A.1 Calculus

To write a definite integral use the *int* command and a subscript and superscript for the limits. The fundamental theorem of calculus can be written as

$$\int_a^b f(x) dx = F(b) - F(a)$$

The Fundamental Theorem of Calculus[6]

If a function f is continuous on the closed interval $[a, b]$ and F is an antiderivative of f on the interval $[a, b]$, then

$$\int_a^b f(x) dx = F(b) - F(a) \quad (10)$$

Problem 1

$$\begin{aligned} \int_0^\pi \sin(\theta) d\theta &= -\cos(\theta) \Big|_0^\pi \\ &= -\cos(\pi) - (-\cos(0)) \\ &= 2 \end{aligned} \quad (11)$$

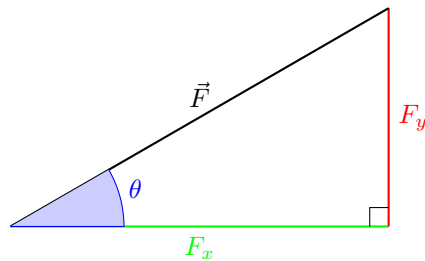
Integration by Parts

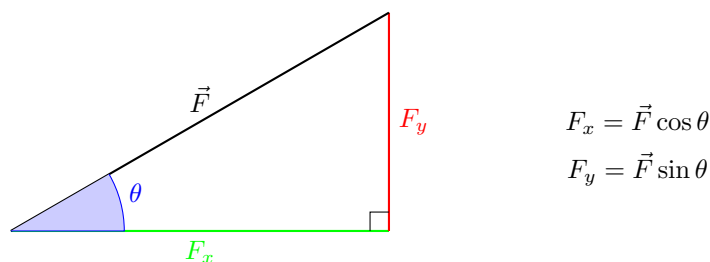
$$\int u dv = uv - \int v du \quad (12)$$

$$\lim_{x \rightarrow \infty} f(x)$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

A.2 Physics





Kinematics Equations in 2-D

$$\langle x(t), y(t) \rangle = \langle x_o + v_{x_o}t + \frac{1}{2}a_x t^2, y_o + v_{y_o}t + \frac{1}{2}a_y t^2 \rangle$$

$$\langle x'(t), y'(t) \rangle = \langle v_{x_o} + a_x t, v_{y_o} + a_y t \rangle$$

$$\langle x''(t), y''(t) \rangle = \langle a_x, a_y \rangle$$

Projectile Motion

Assuming no air resistance the kinematics equations reduce to:

$$\langle x(t), y(t) \rangle = \langle x_o + v_{x_o}t, y_o + v_{y_o}t - \frac{1}{2}gt^2 \rangle$$

$$\langle x'(t), y'(t) \rangle = \langle v_{x_o}, v_{y_o} - gt \rangle \quad (13)$$

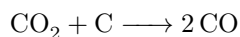
$$\langle x''(t), y''(t) \rangle = \langle 0, -g \rangle$$

A.3 Chemistry

Two packages, *mhchem** and *chemfig*[†], were created specifically for creating chemical formulas and diagrams. Although chemical equations can be typeset in math mode without additional packages the amount of repetitive typing is tedious.

The package *mhchem* uses simple syntax to automate the typesetting of chemical equations. The command *ce* is used to denote a chemical equation and can be used in both text mode and math mode. The code for H₂O is `\ce{H2O}`.[‡]

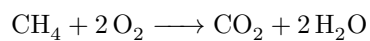
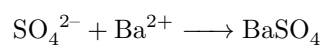
```
\ce{CO2 + C -> 2CO} \\[2ex]
\ce{SO4^2- + Ba^2+ -> BaSO4} \\[2ex]
\ce{CH4 + 2O2 -> CO2 + 2H2O}
```



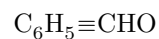
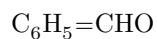
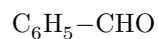
*Documentation for *mhchem* package can be found at <http://mirror.utexas.edu/ctan/macros/latex/contrib/mhchem/mhchem.pdf>

†Documentation for the *chemfig* package can be found at http://ftp.math.purdue.edu/mirrors/ctan.org/macros/generic/chemfig/chemfig_doc_en.pdf

‡Note that the 'H' and 'O' are capitalized; A useful template with examples can be found at <http://www.latextemplates.com/template/chemical-equations>



Bonds can be represented with the *bond* command.



Chemical rings can be created with the chemfig package. A single bond is created with a dash (-), a double bond with an equals sign (=), and a triple bond with a tilde (~). An arbitrary ring with 6 vertices and starting with the letter A is as follows.

```
\chemfig{A*6(=B-C=D~E=F-)}
```

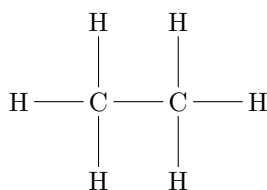
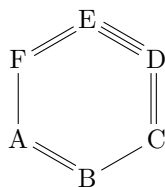


Figure 6: Branched Molecules

A.4 Biology