

# CME 104 L<sup>A</sup>T<sub>E</sub>X starter guide

Enze Chen

Date: July 4, 2018

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	What is LaTeX? . . . . .	3
1.2	Why should I care? . . . . .	3
1.3	How do I use LaTeX? . . . . .	4
1.3.1	Local machine . . . . .	4
1.3.2	Online editor . . . . .	4
<b>2</b>	<b>Basic structure: the .tex file</b>	<b>5</b>
<b>3</b>	<b>General typesetting</b>	<b>6</b>
3.1	Workflow . . . . .	6
3.2	Type and formatting . . . . .	6
3.3	Alignment and Fonts . . . . .	7
3.4	Lists: <code>itemize</code> and <code>enumerate</code> . . . . .	8
<b>4</b>	<b>Math commands/environments</b>	<b>9</b>
4.1	The <code>align</code> environment . . . . .	9
4.2	Vectors and Matrices . . . . .	10
4.3	Derivatives and Integrals . . . . .	10
4.4	Code: the <code>verbatim</code> environment . . . . .	11
<b>5</b>	<b>Figures: the <code>graphicx</code> package</b>	<b>12</b>
<b>6</b>	<b>Tables: the <code>array</code> and <code>booktabs</code> packages</b>	<b>13</b>
<b>7</b>	<b>CME 104: 1-page cheat sheet</b>	<b>14</b>

## Why this document?

LaTeX is something that I've slowly picked up over the years here at Stanford, and now I absolutely love using it. It started as a painstaking chore for CS assignments, but since then I've used it to write research papers, all my problem sets, and even books. I remember thinking early on, "Wow, this LaTeX thingy looks cool, and so many classes advocate its use, but no one seems to want to teach me it!" It's always "Oh, go look online..." which can be helpful once you have a solid foundation and *know what to look for*, but it's pretty daunting for someone just getting started. Therefore, I thought to prepare this as a quick guide to get someone who is [more or less completely] new to LaTeX up and running; essentially, it's everything I wish someone had told me when I first started. Once the basics are there, questions regarding specific use cases are best directed at Google and Stack Overflow, which I credit for much of my personal knowledge of LaTeX and the content presented here.

The first three pages of this document are meant to be motivation (Section 1), an overview of the software, and the basic structure of a `.tex` file (Section 2). If you want to skip to the meatier parts, then Section 3 on typesetting and Section 4 on math commands will be more useful to you. Section 5 and 6 cover the specific syntax surrounding figures and tables respectively. Finally, Section 7 is a concentrated cheat sheet of math commands/symbols that will hopefully serve as a handy reference for CME 104 related work. If you're already familiar with LaTeX, you can jump straight to here.

If you have any questions or comments about this document and what can be added or clarified, please don't hesitate to [reach out](#). Enjoy.

~ Enze

# 1 Introduction

## 1.1 What is LaTeX?

LaTeX, stylized as  $\text{\LaTeX}$  and pronounced either “Lay-tech” or “Lah-tech,” is a document preparation system that uses the TeX typesetting program and various macros to stylize the text. Unlike Microsoft Word and similar What-You-See-Is-What-You-Get (WYSIWYG) word processors, LaTeX relies on a combination of powerful packages and markup conventions to format the plain text you input in a highly structured and customizable way.

## 1.2 Why should I care?

As with everything else in life, before we jump into how to use LaTeX, we consider whether we should be using LaTeX at all—I suppose it is here where I try to compel you into saying “yes.”

### Pros

- Clean and controlled: LaTeX gives users a way to write very complex mathematical formulas and organize content in a way that would otherwise be quite difficult to do in a normal word processor. It has good spacing, sizing, and kerning properties, which applies to regular text as well.
- Professional: LaTeX is used to typeset many scientific documents including journal articles, reports, and books. No matter which domain you work in, LaTeX will give your final product a very professional look and feel, and it’s likely going to come up in your future studies, so might as well get the growing pains out now.
- Cross-referencing: LaTeX can automatically generate a title page, table of contents, bibliography, and index for you that auto-updates as your document changes. With labels, you can also cross-reference tables and figures in the text.
- Portable: Ever received a word document format that you couldn’t open, or it opened with weird symbols? LaTeX not only compiles into a PDF document that will look the same on any platform/operating system, but the same LaTeX file you type now can be compiled on any user’s TeX distribution at any time (provided all packages are installed), providing maximum portability.

### Cons

- Learning curve: Because LaTeX is not WYSIWYG, there is a bit of a learning curve to figure out what the right commands are to achieve a certain format or type a particular symbol. Luckily there are many resources online that will appear with a quick search. Below I will touch upon the most common commands that you’ll probably need, and as with any other learning process, the more you use LaTeX commands the better you’ll remember them!
- Time: Relatedly, typing in LaTeX does take a non-trivial amount of time. I still like to do scratch work and main calculations on paper before typing up my final work in LaTeX (or type as I finish each problem). Depending on the problem set length and your familiarity with LaTeX, expect anywhere from an additional 3-5 hours for problem sets. Obviously the time spent on LaTeX will be worth it for larger projects, but for class assignments this is one trade-off worth considering.

## 1.3 How do I use LaTeX?

LaTeX is freely distributed, and the two main ways people create documents with LaTeX is either on their local machine or using an online editor. For the purposes of CME 104, I will suggest you use an online editor simply to avoid any compatibility issues and/or hassle of set-up, but if you end up liking LaTeX you should look into getting a local distribution (if you already have one, great!).

### 1.3.1 Local machine

You'll need two things to start using LaTeX on your local machine, a **TeX distribution** and any **editor**. Some common ones include:

	Windows	Mac	Linux
TeX	<a href="#">MikTeX</a> or <a href="#">TeX Live</a>	<a href="#">MacTeX</a>	<a href="#">TeX Live</a> or with Linux source
Editor	<a href="#">TeXStudio</a> or <a href="#">TeXMaker</a>	<a href="#">TeXStudio</a> or <a href="#">TeXShop</a>	<a href="#">TeXMaker</a> or Vim/Emacs/Atom

The TeX distribution is in charge of storing all the libraries and packages you need to compile your LaTeX code, and the only difference really is some libraries come with all packages pre-installed while others install packages on-the-fly when you call them for the first time. In either case you might have to update libraries from time to time (no more than once a year).

The choice of editor is largely user preference. Editors that are specifically made for LaTeX might have nice user interfaces, keyboard shortcuts, auto-complete features, and PDF viewing within the software. I recommend any of the above to start, and if a different one suits your fancy, you should use what is most comfortable for you.

### 1.3.2 Online editor

Two popular online editors used nowadays include [Overleaf](#) and [ShareLaTeX](#). There are many reasons to prefer using online editors over local distributions because they will store your files, have most packages available, compile your document as you type, and allow real-time collaboration between you and your peers. Their premium packages cost money, but their free version is enough for most small projects like school work. For this class, if you upload the template for each problem set into your project directory on either of these sites, then everything should compile smoothly to get you started.

## 2 Basic structure: the .tex file

To help you learn the basics of structuring a LaTeX document, I've purposely kept formatting to a minimum on these templates. Every LaTeX document will have a **preamble** and **body**, with each part consisting of the following.

### Preamble

- `\documentclass[options]{doctype}`: This is required so LaTeX knows what format your document takes. *doctype* for you will probably be something like **article**, **report**, or **book**, and we've chosen **article** for problem sets, which is a standard class used for most documents. *options* will differ depending on the document class, and we've kept it simple with just a specification of the font size (**11pt**).
- `\usepackage[options]{packagename}`: Even though your TeX distribution might have every package installed, you still have to specify within each document which package you wish to use before you're allowed to call commands from that package. There are a ton of packages out there (located on [CTAN](#)) and it's a good idea to just stick with the basics here and simply look up what package to include when you need a specific feature. Here are some brief descriptions of what I chose to include in the basic template:
  - **amsmath**, **amssymb**, **amsfonts**: Basic packages attributed to American Mathematical Society (AMS) that provide a bunch of important symbols for math typesetting.
  - **array**, **booktabs**: Basic packages that help with creating tables (see Tables below)
  - **enumerate**: Basic package that makes ordered lists (e.g. 1, 2, 3; see Lists below).
  - **geometry**: Basic package that resizes text area. The *margins* option sets the border width (here, **1in**). We can also specify different *left*, *right*, *top*, and *bottom* margins.
  - **microtype**: Basic package that hyper-optimizes kerning. Not super noticeable differences, but good to include in most instances.
  - **graphicx**: Basic package that helps with creating figures (see Figures below)
  - **mathtools**: Extended package that provides more math symbols. Good to include in most instances; see [documentation](#) for details.
  - **physics**: Extended package that provides shortcut macros for differential equations and various physics operators. See [documentation](#) for details.
- **Macros**: In general, macros are shortcuts that will find and replace every instance of a shortcut command with the user-defined code that follows it. I also like to put other styling commands here. Here are some brief descriptions of what I chose to include in the template:
  - `\setlength`: This is just to help with paragraph spacing, allowing for variable spacing based on how much text is on the page.
  - `\newcommand{\numberthis}`: This is useful for the **align\*** environment (see below) and will automatically insert the correct line number when `\numberthis` is typed. In general, `\newcommand{\command-name}{what-it-does}` gives users a way to define their own macros to simplify typesetting.

### Body

The body of your LaTeX document (i.e. what actually appears to your readers) is everything that is inside the `\begin{document}` and `\end{document}` environment. Here you can type text as normal, and I will go over specific math and text formatting below.

## 3 General typesetting

### 3.1 Workflow

A typical workflow might look something like the following:

- 1) Create a `.tex` file in a directory with the figures you want to use (more on figures later).
- 2) Define `\documentclass` and list all `\usepackage` and macros.
- 3) Start the document body and begin typing, adding extra packages / macros as needed.
- 4) Recompile every couple of lines or when a large formula is typed to *catch any errors early on*. With this step in particular, editors are handy because they can display the PDF and source code side-by-side.

### 3.2 Type and formatting

As previously mentioned, if you just want normal text and paragraphs, you can type as you would in any word processor. Note, however, that you *must* have an extra blank line in order to start a new paragraph, and any extra spaces (1 vs 10) will render the same in LaTeX (1 space). This might be something new to get used to, although you can use it advantageously by making the `.tex` file easier to read. For example, typing

```
$1234 + 56 = 1290 \\ 1 + 3 = 4$
```

and

```
$1234 + 56 = 1290 \\
  1 + 3 =      4$
```

will both appear as

```
1234 + 56 = 1290
1 + 3 = 4
```

but the second option offers greater parallelism and readability *for the writer*. In the next section we'll learn how to make the above equations more readable *for the reader*.

**Environments** allow for customization of a section of text. They start with `\begin{env-name}` and *must be closed* with a corresponding `\end{env-name}`. Pretty much all **commands** in LaTeX start with a backslash (`\`), followed by a keyword, and then curly braces `{ }` enclosing an argument (if one is required). For example, some special fonts and commands that you might find useful are:

- **Bold**, *italics*, and underline can be achieved with `\textbf{word-to-be-bolded}`, `\textit{}`, and `\underline{}`.
- Monospaced computer type can be achieved with the `\texttt{}` command.
- SMALL CAPS (of two different sizes) can be achieved with the `\textsc{}` command.

- **Backslash:** Backslash by itself will create a space, and typing double backslash will create a new line. If you want to actually type the backslash character so it appears to the reader, `\`, use `\textbackslash` in plain text mode or `\backslash` in math mode.
- **Curly braces:** Similarly, because curly braces enclose arguments (and otherwise won't show), if you want curly braces to actually appear, you can force them by preceding each part with a backslash, i.e. typing `"\{\}"` creates `{}`.
- **Quotes:** Open/left quotation marks in LaTeX, “, use two backticks (next to the ‘1’ on a keyboard) and close/right quotation marks use the apostrophe, ”. You don't want to be caught using apostrophes for both; ”ugly!”
- **Spacing and breaks.** The command `\quad` inserts whitespace approximately the width of the letter ‘M,’ and `\qquad` inserts double that width. Use the `\newpage` command to insert a page break.

**Sections:** If you want to split your document into numbered sections, use the `\section{Header name}` command. Subsections are specified with `\subsection{}` and `\subsubsection{}`, and are automatically numbered in order. If at any point you want a section header *without* a number, include an asterisk like `\section*{Header name}` (many numbered options in LaTeX can be removed with the addition of an asterisk).

**Cross-referencing:** LaTeX also supports internal references to different parts of a document. Anything that is numbered (e.g. sections, equations, figures, tables) can have a `\label{labelname}` placed by it, and this can be referenced elsewhere in the text using `\ref{labelname}`. When you do this, I would advise informative names that differ by object, for example, `\label{fig:wave}` or `\label{sec:LU}` for figures and sections respectively.

**Comments:** As shown in the template, the percent sign `%` will comment out the rest of the line. Blocks of comments can be done using the `\begin{comment}` environment.

### 3.3 Alignment and Fonts

You will notice that the homework templates begin with the `\begin{center}` environment to center the title and names. Everything typed inside this environment will be centered and everything outside returns to the default. LaTeX, like most other software, is left-aligned by default, but if you would like to right-align text, you can use the `\begin{flushright}` environment.

Sometimes, you might want text to be aligned left and right *in the same line*, like when constructing headers [for a resume]. You can accomplish this using the `\hfill` command, which pushes everything after it to the right (it essentially fills the space in between with whitespace).

Absolute font size is controlled in the `documentclass` declaration, but you can also specify relative font size by enclosing sections of text in curly braces, as shown in the template. Choices include `\tiny`, `\footnotesize`, `\small`, `\large`, `\Large`, `\LARGE`, `\huge`, and `\HUGE`.

You probably will only need font color for specific words and phrases here or there, and for that there is the `\textcolor{color}{text}` command such as `\textcolor{red}{this}` → **this**.

### 3.4 Lists: itemize and enumerate

If you would like to create lists of items, the two ways are using the `itemize` and `enumerate` environments. Both use `\item` to start a new entry, and while the former creates bullet points, the second one creates ordered lists that allow you to specify the style. If you type

```
\begin{itemize}
  \item One fish
  \item Two fish
\end{itemize}
\begin{enumerate}[a)]
  \item Red fish
  \item Blue fish
\end{enumerate}
```

what you will get looks like

- One fish
  - Two fish
- a) Red fish
- b) Blue fish

where in the `enumerate` environment the `[a)]` option creates the list bullet style. Other style options for lists include `[1.]`, `[(i)]`, `[a.]`, etc.

Now, of course, I herd you like lists, so I put a list in your list so you can list while you list<sup>1</sup>:

1. Ordered list item 1 contains
  - Unordered list item 1
  - Unordered list item 2 contains
    - i. Inner ordered list item
2. Ordered list item 2

```
\begin{enumerate}[1.]
  \item Ordered list item 1 contains
  \begin{itemize}
    \item Unordered list item 1
    \item Unordered list item 2 contains
    \begin{enumerate}[i.]
      \item Inner ordered list item
    \end{enumerate}
  \end{itemize}
  \item Ordered list item 2
\end{enumerate}
```

---

<sup>1</sup>[Xzibit Yo Dawg meme](#). Footnotes can be made with the `\footnote{}` command.



## 4 Math commands/environments

Time for the fun part! Math environments in LaTeX can be created in a variety of ways, the most typical one being the dollar sign \$. For example, if you type

```
$3 \times 2 - 1 = 5$
```

you will get  $3 \times 2 - 1 = 5$  inline with the text. Anything enclosed with dollar signs will render in **math mode**, including text, *which will unfortunately look like this*, so if you need text within dollar signs, use the `\text{}` command. Just as with curly braces, if you ever need to type the actual dollar sign (such as with monetary figures), add a backslash before it, ‘\\$.’ There are too many special math commands for me to list here, so please consult the 1-page cheat sheet or online for common math symbols (e.g. how to type fractions, integrals, etc.).

Another way to trigger the math environment is to use ‘\[’ and ‘\].’ Anything within these special brackets will not only render in math mode, but it will do so centered on a new line, such as

$$3 \times 2 - \frac{2}{2} = 5.$$

With the exception of small inline expressions, I *much prefer* to use square brackets over dollar signs when typing math because of the clarity and emphasis. Also, tall expressions like fractions and summations will get compressed to “fit” inline when using dollar signs, e.g.  $\frac{1}{2}$  and  $\sum_1^5$ , but they will fully expand in the bracket environment.

### 4.1 The align environment

Yet another popular and extremely useful way to trigger the math environment is using the `\begin{align}` environment, which allows you to type multiple lines of math, each separated with two backslashes, `\\`, and aligned using the ampersand character `&`. For example, to get

$$c^2 = 4 \left( \frac{ab}{2} \right) + (b - a)^2 \tag{1}$$

$$= 2ab + b^2 - 2ab + a^2 \tag{2}$$

$$= a^2 + b^2 \tag{3}$$

one can type

```
\begin{align}
c^2 &= 4 \left( \frac{ab}{2} \right) + (b - a)^2 \\
&= 2ab + b^2 - 2ab + a^2 \\
&= a^2 + b^2
\end{align}
```

Note how in each line the ampersand lines up the equal signs, and how each line ends with two backslashes so LaTeX knows to begin the next line. Each line must have the same number of `&` characters. `align` will auto-center the expressions as well. Again, we don’t actually have to create a new line for LaTeX to render (all of that could be typed onto a single line), but we do so for readability. If we want an aligned environment *without* numbering each equation, we can use the asterisk as before, `\begin{align*}` and corresponding `\end{align*}`.

## 4.2 Vectors and Matrices

We will be handling vectors and matrices a lot in this class. Vectors are most clearly specified with an arrow above a lowercase letter using the `\vec{u}` command in math mode, e.g.  $\vec{u}$ . Matrices are represented with a capital letter which can be directly typed in math mode or bolded, i.e.  $A$  with ‘`$A$`’ or  $\mathbf{A}$  with ‘`\textbf{A}`’ will work, but not plain-text  $A$ .

Explicitly writing the entries of a matrix can be done with either the `\begin{matrix}` (no side bars), `\begin{bmatrix}` (square brackets), `\begin{pmatrix}` (parentheses), or `\begin{vmatrix}` (vertical bars) environments. Following the previous convention, use ampersands `&` to separate elements in each row and double backslash “`\\`” to separate different rows. Each `matrix` environment can be sized without explicit pre-specification, but once the first row is specified, all subsequent rows must have the same number of elements / columns or it will throw an error. Examples include

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
\[ A = \begin{pmatrix}
  1 & 2 & 3 \\
  4 & 5 & 6
\end{pmatrix}, \quad \qquad
I = \begin{bmatrix}
  1 & 0 \\
  0 & 1
\end{bmatrix} \]
```

Vectors are just 1-dimensional matrices and constructed as such, namely  $\vec{p} = \begin{bmatrix} 1 & 2 \end{bmatrix}$  or  $\vec{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ . To save space and for presentation purposes, you might see a column vector represented as  $\vec{u} = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$ , written as the transpose of a row vector. Do remember, however, that *by convention vectors are assumed to be column vectors*.

## 4.3 Derivatives and Integrals

There are many packages out there that simplify the work of typesetting derivatives, but I have found `physics` to be nice, and so I’ve included it in the template. To obtain an ordinary derivative, simply use the macro `\dv[order; default = 1]{optional top argument}{required bottom argument}`. Concrete examples include

$$\frac{d}{dt}, \quad \frac{d^2 T}{dx^2}, \quad \frac{d^4}{dx^4}$$

```
\[ \dv{t}, \quad \qquad
\dv[2]{T}{x}, \quad \qquad
\dv[4]{x} \]
```

Partial derivatives follow the same syntax but use the `\pdv` macro. It’s pretty nifty!

The symbol for gradients is the **nabla** symbol, which is written using `\nabla` in math mode,  $\nabla$ . Divergence is nabla followed by a centered dot,  $\nabla \cdot$  (`\nabla \cdot`), and this dot is also the preferred symbol for multiplication (opposed to asterisk;  $3 \cdot 2$  vs.  $3 * 2$ ). Laplacians are commonly written as  $\nabla^2$ , `\nabla^2`, or  $\Delta$ , which is `\triangle`.

Integrals take the macro `\int_{lower limit}^{upper limit}` and end with `\dd{var}`, also thanks to the `physics` package. An example would be

$$\int_0^\infty \frac{\sin(x)}{x} dx$$

```
\[ \int_0^{\infty} \frac{\sin(x)}{x} \dd{x} \]
```

Summations and products work in a similar fashion with the `\sum` and `\prod` commands respectively.

$$\sum_{i=1}^{\infty} \frac{1}{i} \quad \prod_{i=1}^n i$$

```
\[ \sum_{i = 1}^{\infty} \frac{1}{i} \quad \prod_{i = 1}^n i \]
```

### Note about {}

You may have noticed that in the code above (and other places), we used curly braces to enclose large quantities that we want to put into a subscript, exponent, fraction, etc. It's important to recognize that LaTeX will not capture more than one character for an operator, i.e. typing `A^-23` results in  $A^{-23}$  (only the minus sign is in the exponent) whereas typing `A^{-23}` is probably what you want,  $A^{-23}$ . You will need braces whenever you want an operation or command to be applied to a long expression.

## 4.4 Code: the verbatim environment

Now, there are many times where we actually *want* the reader to see the raw math type without it compiling, such as the many places above, and more generally when you want to include code (like MATLAB!) without LaTeX freaking out. This is accomplished in a very basic manner with the `\begin{verbatim}` environment, which eliminates all formatting and indentation and spits out the text exactly as you type it. You should use this environment if you're attaching the MATLAB code into your LaTeX writeup, and for readability, **please add two spaces for each level of indentation** such as for loops and if statements (unfortunately, tabs don't always work well in `verbatim`).

`listings` is another package that works very well, particularly with its [numerous customizations](#). `minted` is another package that [looks quite nice](#), but I've personally had installation issues.

## 5 Figures: the `graphicx` package

LaTeX is great for many things, including making its own vectorized graphics (see [TikZ](#) documentation), but sometimes we want to attach a figure created by another program. The `graphicx` package is extremely useful for this task, as it allows us to type the following:

```
\begin{figure}[!h]
  \centering
  \includegraphics[width=0.2\linewidth]{stanford.png}
  \caption{Stanford emblem.}
  \label{fig:stan}
\end{figure}
```

to produce:



Figure 1: Stanford emblem.

Let’s dissect each piece. The `\begin{figure}` environment is what we call for figures. The `[!h]` option is a handy flag that optimizes placing of the figure below/near where we specify, or else (perhaps oddly) figures will float to the top of every page they appear on.

`\centering` is an alignment command that centers the figure within the frame, and `\includegraphics` grabs the actual figure from its source file. The `width` option is handy for controlling figure size, and most MATLAB plots I generate I usually give between 0.4 – 0.5 of the total `\linewidth`. Inside the braces we write the name of the figure file (with explicit extension to be safe), and PNG, JPG, and PDF are all supported file types.

`\caption{}` is quite useful as you can explain a little bit about the figure, and `\label{}` allows cross-referencing of Figure 1 within the text (I typed ‘Figure `\ref{fig:stan}`’).

**Note on file path:** The place LaTeX looks by default is the current folder that the `.tex` file is in. If you have your figures organized elsewhere, say in a subdirectory called “fig” in the current directory (often a good idea for larger projects!), you can add a line `\graphicspath{{./fig/}}` in the preamble to tell LaTeX where to look. In general, “./fig/” can be replaced by any relative or absolute path.

## 6 Tables: the array and booktabs packages

LaTeX offers a great deal of flexibility in creating and customizing tables. The `array` and `booktabs` packages really make this all possible, so they should always be included. We already saw an example of a table up above, and here I'll give another minimum working example.

```
\begin{table}[!h]
  \centering
  \begin{tabular}{l|cc}
    \ & A & B \\
    1 & Yes & No \\
    2 &  $\frac{2}{3}$  &  $-0.4$ 
  \end{tabular}
  \caption{My table.}
  \label{tab:example}
\end{table}
```

	A	B
1	Yes	No
2	$\frac{2}{3}$	$-0.4$

Table 1: My table.

As you can see, tables and figures share a lot of similarities. We use the `\begin{table}` environment with the appropriate alignment options. Inside, we call another `\begin{tabular}` environment, which takes as options an **explicit, pre-specified** number of columns (unlike implicit for matrices) with **explicit** alignment, either left (l), center (c), or right (r), that can differ for each column. Vertical lines, made using the key above the return key on the keyboard, determine whether or not there is a line separating the two columns.

The body of a table is very similar to that of a matrix. Again, ampersands separate the columns and double backslashes separate the rows. The number of columns in each row must match what is specified above, and while empty elements can be left blank (i.e. we could have started row one with just an ampersand), I like to include a backslash to explicitly signal that the cell is empty. `booktabs` comes with the `\toprule`, `\midrule`, and `\bottomrule` commands that draw horizontal lines separating the rows of the table. Note that `\toprule` and `\bottomrule` are slightly thicker than `\midrule`. By default, tables **are not** in math mode, so any cells with math content need dollar signs *for each cell*. Captions and labels work similar to that of figures.

## 7 CME 104: 1-page cheat sheet

- Bold: `\textbf{A}`, e.g. **A**.
  - Italics: `\textit{A}`, *A*.
  - Underline: `\underline{A}`, A.
  - Monospace: `\texttt{A}`, A.
  - Small caps: `\textsc{a}`, A.
  - Normal (in math mode (mm)): `\text{}`.
  - Emphasis: `\emph{A}`, *A*.
  - Color: `\textcolor{color}{text}`, **B**.
  - Blackboard (mm): `\mathbb{Z}`,  $\mathbb{Z}$ .
  - Calligraphy (mm): `\mathcal{T}`,  $\mathcal{T}$ .
  - Fraktur (mm): `\mathfrak{R}`,  $\mathfrak{R}$ .
  - Greek (mm): `\alpha`,  $\alpha$ . `\beta`,  $\beta$ . `\pi`,  $\pi$ . `\psi`,  $\psi$ . `\Theta`,  $\Theta$ . And many more!
  - Infinity (mm): `\infty`,  $\infty$ .
- 
- Math mode: `\math{}` or `\[math]\]`.
  - Align: `\begin{align}` or `\begin{align*}`.
  - Force tall expression: `\displaystyle\{com\}`.
- 
- Degree: `35^\circ`,  $35^\circ$ .
  - Multiplication: `\cdot`,  $\cdot$ . `\times`,  $\times$ .
  - Fraction: `\frac{a}{b}`,  $\frac{a}{b}$ .
  - Square root: `\sqrt{x}`,  $\sqrt{x}$ .
  - *n*th root: `\sqrt[n]{y}`,  $\sqrt[n]{y}$ .
  - Combination: `\binom{n}{r}`,  $\binom{n}{r}$ .
  - Vector: `\vec{p}`,  $\vec{p}$ .
  - Norm: `\|p\|_2`,  $\|p\|_2$ . `\abs{a}`,  $|a|$ .
  - Matrix: `\begin{bmatrix} 1 & 0 \\ \end{bmatrix}`.
  - Markings: `\hat{p}`,  $\hat{p}$ . `\tilde{0}`,  $\tilde{0}$ . `\bar{x}`,  $\bar{x}$ . `\dot{u}`,  $\dot{u}$ .
  - Braces: `\overbrace{text}^p`,  $\overbrace{text}^p$ . `\underbrace{p}_{text}`,  $\underbrace{p}_{text}$ .
  - Trigonometry: `\sin`,  $\sin$ . `\cos`,  $\cos$ . `\arctan`,  $\arctan$ . `\sinh`,  $\sinh$ . etc.
  - Functions: `\exp`,  $\exp$ . `\log`,  $\log$ . `\ln`,  $\ln$ .
- 
- Limits: `\lim`,  $\lim$ . `\max`,  $\max$ . `\min`,  $\min$ .
  - Gradient: `\nabla`,  $\nabla$ . `\triangle`,  $\triangle$ .
  - Summation: `\sum_{lower}^{upper}`,  $\sum_{i=0}^k$ .
  - Integral: `\int_{l}^u dd{}`,  $\int_0^L dx$ .
  - Product: `\prod_{lower}^{upper}`,  $\prod_{i=1}^n$ .
  - ODE: `\dv[order]{top}{bottom}`,  $\frac{d^2F}{dx^2}$ .
  - PDE: `\pdv[order]{top}{bottom}`,  $\frac{\partial G}{\partial y}$ .
- 
- Comparison: `\le`,  $\leq$ . `\ge`,  $\geq$ . `\neq`,  $\neq$ .
  - Similarity: `\sim`,  $\sim$ . `\approx`,  $\approx$ . `\cong`,  $\cong$ . `\equiv`,  $\equiv$ . `\perp`,  $\perp$ . `\parallel`,  $\parallel$ .
  - Sets: `\{ \mid, \in \}`,  $\{x \mid x \in \mathbb{N}\}$ . `\subseteq`,  $\subseteq$ .
  - Empty set: `\emptyset`,  $\emptyset$ . `\varnothing`,  $\varnothing$ .
  - Membership: `\forall`,  $\forall$ . `\exists`,  $\exists$ .
  - Logic: `\vee`,  $\vee$ . `\wedge`,  $\wedge$ . `\neg`,  $\neg$ .
  - Square: `\square`,  $\square$ . `\blacksquare`,  $\blacksquare$ .
  - Signs: `\pm`,  $\pm$ . `\mp`,  $\mp$ .
  - Dots: `\cdots`,  $\cdots$ . `\dots`,  $\dots$ . `\vdots`,  $\vdots$ .
  - Parentheses: `\left(`,  $\left($ . `\right)`,  $\right)$ .
  - Brackets: `\left[`,  $\left[$ . `\right]`,  $\right]$ .
  - Arrows: `\rightarrow`,  $\rightarrow$ . `\Leftarrow`,  $\Leftarrow$ . `\longleftrightarrow`,  $\longleftrightarrow$ .
- 
- ```

\begin{figure}[!h]
\centering
\includegraphics[width=0.4\linewidth]{myfig.png}
\caption{My figure.}
\label{fig:myfig}
\end{figure}

\begin{table}[!h]
\centering
\begin{tabular}{l|c}
\hline
& A \\ \hline
foo & bar
\end{tabular}
\caption{My table.}
\label{tab:mytab}
\end{table}

```