Learning LaTeX with MEGC

University of Michigan
Ann Arbor, Michigan

January 23rd, 2020

1 Introduction

1.1 Editors

There are many different LaTeX editors out there. This website gives a good overview of the different editors, their benefits and disadvantages, and what operating systems they work on:

`http://en.wikipedia.org/wiki/Comparison_of_TeX_editors`.

It is important to note that you can do all of your LaTeX using online editors like ShareLaTeX, but you need an internet connection to do so. If you wish to work on your local machine (which is typically faster to compile than going through a cloud service) you may want to consider downloading an editor. When you download, you will receive a default editor called TeXworks. This is a bare bones editor and you could greatly benefit from an alternative. Some editors that MEGC members have used or have considered are the following:

- **Windows:** `https://www.youtube.com/watch?v=g6ez7sbaiWc`
  - TeXstudio ← *This is T.J.'s favorite editor*
  - Texmaker ← *This is Jeff's favorite editor*
  - TeXnicCenter ← *This editor is not our favorite, but is the most widely used on Windows OS*
- **Mac OS:** `https://www.youtube.com/watch?v=5CNmIaRxS20`
  - TeXShop ← *Sara's go to editor on Mac*
  - Latexian ← *Does live previewing but somewhat out of date*
- **Ubuntu (Linux):** `https://www.youtube.com/watch?v=g6ez7sbaiWc`
  - Texmaker
  - Gummi ← *Brandon's go to editor in Ubuntu and has live previewing*
  - Kile ← *Sara's go to editor in Ubuntu*
  - AUCTeX ← *Works with Emacs*

1.2 Syntax

- Command Syntax
  - For the most part, every command in LaTeX can be described as `\command[options]{argument}`.
- Environments
  - Writing in LaTeX can be thought of as writing in multiple different environments. You can have text environments, math environments, table environments, figure environments, and many many more. These environments can be nested and are typically marked by `\begin{·}` and `\end{·}` where what is inside of {·} denotes the environment. By default you are usually in the text environment when writing a document.
  - One exception to the `\begin{·}` and `\end{·}` markings is when you switch between text and math environments while keeping the math inline with the text. This switch is marked by `$·$`, but we will get to more of this in Section 3.

1.3 Components to a LaTeX Document

- Preamble
  - The preamble sets up the document by telling LaTeX what document type you are working on, what the page layout should look like, what packages you plan to use, and allows you to define new commands. The preamble is everything before the `\begin{document}` command.
- Content
  - The content of the LaTeX document is everything between the `\begin{document}` command and the `\end{document}` command. This includes everything that you see in the finalized PDF document. By default the `document` environment is a text environment. Everything you write between `\begin{document}` and `\end{document}` will be plain text unless an alternative environment is indicated.

1.4 Document Class

There are many different classes of documents you can write in LaTeX. The document class can be thought of the style that you are writing in. The document class is defined at the top of the preamble (line 6 in this .tex file) and is generally denoted by `\documentclass[options]{class}`.

- `article` - for short documents such as a publication (this document)
- `report` - for longer technical documents; like articles, but has chapters
- `book` - for large documents, such as books
- `letter` - for writing letters (has other special properties which have to be set)

A full list of document classes and class options can be found here: `https://en.wikibooks.org/wiki/LaTeX/Document_Structure#Document_classes`.

1.5 Packages

Packages are essentially libraries of commands that you are telling LaTeX you will be using throughout the document. Packages may be needed for specific mathematical symbols, fonts, colors, the ability to insert figures, the ability to change document formatting, and much much more. An example of a package used in this document is the following:

```
\usepackage[margin = 2cm]{geometry}
```

The package `geometry` allows for an easy way to customize the document margins. The option `[margin = 2cm]` allows for a 2 centimeter margin all the way around the page without including the header or footer in that cutoff (ie the header and footer are within 2 centimeters of the paper's edge). We could change the option to `[margin = 2cm, includefoot, includehead]` to make sure that the header and footer are outside of this margin cutoff as well. The option `margin` can also be changed to `top`, `bottom`, `left`, and `right`. You can always look up a package name to figure out the specific options for that given package. Take a minute to try changing the options on the `geometry` package in the preamble to see how it effects this document.

2 General Writing with LaTeX

In this section, we are mainly going to talk about general writing in LaTeX. To write in LaTeX, you can just start typing regularly like you would do in a normal text editor (e.g. Notepad in Windows; TextEdit in Mac). However, your document will be void of any formatting. Therefore, to make it easier to read, we must add in different types of formatting.

Note: We suggest that you keep every sentence on its own separate line when writing. This way if your code fails to compile, you should get an error message saying which line caused the crash and you can easily debug the problem.

2.1 General Formatting

- Basic text formatting
  - For basic text formatting, we can include *italics* with `\textit{·}`, underline with `\underline{·}`, or **bold face** with `\textit{·}`. You can format text with ***all three*** by nesting the commands!
  - The percent symbol (%) is used to comment any lines out
- Text color
  - You can change the color of your text. For example, we can make our text red, blue, maize, dark grey, etc. To change the colors of text though you must make sure you have the `color` package included in your preamble.
  - Information about the different colors, the packages needed, and how to create your own color can be found here: `http://en.wikibooks.org/wiki/LaTeX/Colors`.
- Change font type
  - You can easily change the font size in LaTeX using commands.
    - For example this text is tiny
    - For example this text is Large

    We are not going to show you how to do much font changes in this presentation, as it is not the most vital. You can easily search online for how to do this (for a start, look here: `http://en.wikibooks.org/wiki/LaTeX/Fonts`.
- Starting new paragraph
  - Leave a blank line between sentences ← *Would recommend most*
  - Use the command `\newline` followed by `\indent` ← *Would avoid to keep .tex readable*
  - Use the command `\\` ← *Would avoid to keep .tex readable*

2.2 Creating List Environments

LaTeX has different kinds of list environments, and you can even create your own (e.g. see the `Resume_Template` that we will give you at the end of the workshop). You can read more about the lists here: `http://en.wikibooks.org/wiki/LaTeX/List_Structures`. In general, the three standard list environments are the following:

1. **Enumerate** – What was used for this list.
2. **Itemize** – Standard bulleted item list; can choose your own bullets and place them in brackets right after item. For example `\item[$\circ$]` will give you this symbol ∘ as your bullet.
3. **Description** – More freedom to specify the item label.

For a list, you must start with for example `\begin{enumerate}` and end with `\end{enumerate}` (or `\begin{itemize}`; `\end{itemize}`, etc.). In between those two commands, you can have as many items as you would like. You must signify each item using the command `\item`.

2.3 Labels and References

One of the best parts of LaTeX is the ease at which you can reference the different elements in your document, and LaTeX will automatically number everything accordingly (if you have different preferences for numbering, you can make your own styling–but that is a little more advanced). Labels and references are commonly used for these elements:

- Sections in the document (e.g. a numbered subsection, a chapter, an appendix)
- Figure
- Equations
- Table
- Citation

Now look at the LaTeX code and see what was typed just below `\subsection{Labels and References}`. You should see a `\label{sec:labelsreferences}`. This command assigned the label "sec:labelsreference" to this very subsection. Now if you want to refer to this section anywhere throughout the document, all you have to do is type the command `\ref{sec:labelsreferences}` and LaTeX will reference Section 2.3. There is a special reference command you can use for equations that will include parenthesis around the citation. The command is `\eqref{·}`.

The same labeling technique can be used for equations, figures, tables, appendices, chapters etc. It is usually considered good practice to put a signifier in your label as to what you are actually labeling. For example, if you were to label an equation, such as the first equation in Section 3.2, you could call it "eq:...". This labeling convention just makes things easy to interpret when writing. So for the first equation in Section 3.2, we have labeled the equation `\label{eq:example}`. Now we can refer to Eqn. (1) by using the command `\eqref{eq:example}`.

The signifier for a table is usually `tab:`, the signifier for a figure is usually `fig:`, and the signifier for an equation is usually `eq:`. You can then create your own signifiers for everything else. We will discuss how to make references to citations in Section 7, but just as a heads up, instead of using the command `\ref{·}` we will use `\cite{·}` for bibliography citations.

Practice your text skills by trying to replicate Exercise1.pdf

3 Math environment

Another wonderful aspect of LaTeX is its ability to seamlessly move from mathematical environments to text environments. When we are going into math environment, we are telling LaTeX that we are now using math symbols. There are two ways to enter the math environment:

1. Inline with your other text. You enter in and out of the math environment using a set of dollar signs: `$math stuff$`.
2. In a display style. For the display style you entering into a different environment (ie. equation, align, matrix, etc.) which assumes everything you are typing is a math symbol unless you specify otherwise. You will see more of this in Section 3.2.

Some other notes for the math environment:

- Spacing is done automatically unless forced with dedicated commands (`\quad`, `\vspace`, `\hspace`, etc...these are described in detail in Section 5)
- Empty lines are not allowed so use comments (%...) to clean things up.
- After entering a math environment, math fonts will be used unless the `\text{·}` environment is invoked.
- As was stated above, to use the math environment in prose writing, you must separate whatever you are writing that is "math" by dollar signs (i.e. `$math stuff$`). Thus, if you want to write *sigma* in line with your text, you should type `$\sigma$` to get σ.

3.1 Basic Math Commands

Some basic math environment commands:

- `\\` is used to end a line and start a new one
- The **ampersand** (&) is usually used for alignment in the math environment. You will see this when we create arrays and equations with more than one line.
- `_` gives you a subscript. For example to get x_2, you type `$x_2$`.
- `^` gives you a superscript. For example to get x^2, you type `$x^2$`.
- {·} are used to include extra but necessary information for a command. For example, if you want to type x_{i+1}, you must put `i+1` inside of {·}, otherwise the subscript only reads the `i`. So to get x_{i+1} you would type `$x_{i+1}$`. Without the {·} around the `i+1`, it looks like $x_i + 1$.
- **Greek symbols** are widely used in the math mode. Greek symbols are usually the name of the symbol with a \ at the beginning. For example, δ is typed as `$\delta$`. For the upper-case version of the greek symbols, you usually just capitalize the first letter of the greek symbol command. Thus, for Δ, we type `$\Delta$`. For more on Greek symbols, look here: `http://en.wikibooks.org/wiki/LaTeX/Mathematics#List_of_Mathematical_Symbols`.
- A **fraction** command is another widely used math command. For a fraction, you will use the command `\frac{}{}$` where you must input two arguments. The first argument is the *numerator* and the second argument is the *denominator*. So for example, if you want $\frac{1}{2}$, you would type it as `$\frac{1}{2}$`.

3.2 Equations

One of the most attractive parts of LaTeX is the ease and cleanliness when typing complicated equations. Here is our basic equation presented using the align environment:

$$\frac{\partial u}{\partial y} + C_1^2 \tag{1}$$

The align environment is the same as the equation environment shown below however we recommend always using align because it allows for added features:

$$\frac{\partial u}{\partial y} + C_1^2 \tag{2}$$

Adding a star inside of the brackets of `equation` (or any numbered environment for that matter) will take away the numbering associated with it. For example, if we begin our `equation` environment with `\begin{environment*}` we get the following:

$$\frac{\partial u}{\partial y} + C_1^2$$

Using the `align` command, we can easily align parts of a multi-lined equation using `&` as the alignment object for each line (where you end a line of the equation using the standard `\\`):

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

Let's show another example of using alignment with our equations. Let $y = 3$ then we can solve for x:

$$\begin{aligned} x &= y^3 + e^{y^2} - y + \log y \\ &= (3)^3 + e^{3^2} - 3 + \log 3 \\ &= 27 + e^9 - 3 + \log 3 \\ &= 30 + e^9 - \log 3 \\ &\approx 30 + 8103.082 - 1.099 \\ &\approx \mathbf{8131.983} \end{aligned}$$

We can also implement more complicated math symbols such as summation and integration. To demonstrate this, we will write out the definition to a definite integral:

Definition 1. Let $f(x)$ be a function which is continuous on the closed interval $[a, b]$. The definite integral of $f(x)$ from a to b is

$$\int_a^b f(x)\,dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i)\Delta x$$

where

$$\sum_{i=1}^{n} \Delta x_i$$

is a Riemann sum of $f(x)$ on $[a, b]$.

It should be noted that using `,\` right after `f(x)` in the integral, gives the correct spacing between $f(x)$ and dx.

You may also need to utilize the `\displaystyle` command when your integrals or summations are inline. So for example:

$$f(x) = \frac{\int_0^1 g(x)\,dx}{\int_0^1 h(x)\,dx} \tag{3}$$

Here is the same function but written using `\displaystyle` before `\int`:

$$f(x) = \frac{\displaystyle\int_0^1 g(x)\,dx}{\displaystyle\int_0^1 h(x)\,dx} \tag{4}$$

If you ever need to write an integral, a fraction, or a summation inline, you can use `\displaystyle` command.

3.3 Matrices

We can make a matrix by entering the `matrix` environment. Here, we use `&` to separate items in the row of a matrix and we use \\ to create new rows. Here is the basic 3×3 identity matrix using the matrix environment `\begin{matrix}` and `\end{matrix}`:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

You can add brackets around it by typing `\left[` and `\right]` before and after you enter the `matrix` environment, respectively:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

You can also do the same thing as above but using the `array` environment. This environment is a little more complicated than the matrix environment, but does give you more control when you get the hang of it. You can learn more about it here: `http://en.wikibooks.org/wiki/LaTeX/Mathematics#Matrices_and_arrays`. Here is an example of the same thing, but using the `array` environment:

$$\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

We can also use the `array` environment for certain types of equations:

$$F(x) = \begin{cases} 0, & x \leq 0 \\ \frac{\partial u}{\partial y} + C_1^2 & x > 0 \end{cases} \tag{5}$$

Note here that using the command `\left{` puts a curly bracket to encompass the left side of the array, but when using `\left`, you must also have a `\right` command at some point. However, since we do not want a curly bracket on the right, we use `\right.` (`\right` with a period). Using the `\dfrac{numerator}{denominator}` command will make the fractions larger and easier to read

$$F(x) = \begin{cases} 0, & x \leq 0 \\ \dfrac{\partial u}{\partial y} + C_1^2 & x > 0 \end{cases} \tag{6}$$

Practice your math skills by attempting to replicate Exercise2.pdf

4 Tables and Figures

LaTeX has the ability to include nicely formatted tables and figures. These can sometimes be a pain to get in the right place because they are known in LaTeX as floats. LaTeX typically places floats where it deems best for them to go, but this typically isn't where you want them placed. You can constrain a float to be in the location it is typed in the code using the `[h]` command directly after the begin environment command (see code for the following table and figure for an example) but there are many other ways to place floats where you want them. A good description on how to control floats is listed here: `http://tex.stackexchange.com/questions/39017/how-to-influence-the-position-of-float-environments-like-figure-and-table-in-lat`.

4.1 Tables

To create tables, we use the `tabular` environment nested within a `table` environment. It is important to note that most tables will begin with `\begin{tabular}{·}` and what is inside of the {·} helps preallocate the table itself. Take a look at the following table.

Greek symbol	**Upper-case**	**Lower-case**
Omega	`\Omega` (Ω)	`\omega` (ω)
Beta	N/A	`\beta` (β)
Delta	`\Delta` (Δ)	`\delta` (δ)
Sigma	`\Sigma` (Σ)	`\sigma` (σ)
Alpha	N/A	`\alpha` (α)

Table 1: Some Greek symbols and their corresponding commands.

This table has three columns that are all left aligned. If you take a look at the LaTeX code that generated the table you will see that it begins with `\begin{tabular}{l | l | l}`. The `l`'s in this preallocation define the alignment of each column. Preallocation can be defined with `l`'s, `c`'s, or `r`'s for alignment. The `|`'s that separates each `l` tell LaTeX to put a vertical line between each column. You should also notice that each row uses ampersands (`&`) to signify which content belongs in each column and \\ are used to signify when to start a new row (just like when creating matrices as explained in Section 3.3). Additionally, the command `\hline` can be used to put in horizontal lines where needed. A really good overview of LaTeX tables can be seen here: `http://en.wikibooks.org/wiki/LaTeX/Tables`.

4.2 Figures

In order to insert figures into your LaTeX documents you must include the `graphicx` package. To begin a figure you will enter the `figure` environment. Once in the `figure` environment you can then include a figure using the `\includegraphics{·}` command where the figure name (or figure path relative to your `.tex` file location and filename) are in the {·}. A common method of sizing an image is to size it relative to the paper's text width. An example of this is below.

Figure 1: Picture of an awesome cat.

If you look at the LaTeX code that generated this figure you will see that the figure width was set to $\frac{1}{2}$ the documents text width. Additionally a caption can be added to a figure using the `\caption{·}` command where the caption text is between the {·}. More information on figures can be found here: `https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions`.

LaTeX also has the ability for you to draw your own pictures. We are not going to go into that for this presentation. But if you want to learn more about it, you can find out more here: `http://en.wikibooks.org/wiki/LaTeX/PGF/TikZ`

Practice your figure and table skills by trying to replicate Exercise3.pdf

5 Space Management

A great freedom (and curse) of LaTeX is that you have free range to change mostly any spacing on the page. Sometimes you need to nudge something up, down, left, or right. The following section will discuss how to do this.

5.1 Units

Common units used in LaTeX:

- pt - point (1 in = 72.27 pt)
- in - inch (1 in = 25.4 mm)
- cm - centimeter (1 cm = 10 mm)
- mm - millimeter
- em - roughly the width of an 'M' (uppercase) in the current font
- ex - roughly the height of an 'x' in the current font

Useful LaTeX measurements to reference:

- `\textwidth` - The width of the text on the page (margin to margin)
- `\textheight` - The height of the text on the page (margin to margin)

5.2 Creating Whitespace

The following are commonly used commands for creating whitespace.

- `\noindent` - Removes indent at the beginning of a new paragraph
- `\` - About a single spacebar of spacing
- `\quad` - About the spacing of 'M'
- `\qquad` - Add the spacing of 'MM'
- `\vspace{·}` and `vspace*{·}` - Add vertical space of {·} units(*indicates manual override)
- `\hspace{·}` and `hspace*{·}` - Add horizontal space of {·} units (*indicates manual override)
- `\vfill` - Adds whitespace to fill the page vertically
- `\hfill` - Adds whitespace to fill the page horizontally

5.3 Minipages

Sometimes you want to section out a page into subsections of space making it easier to manipulate sets of objects or place objects relative to one another. This can be difficult to do traditionally in LaTeX but minipages can be used to make it easier. Minipages is somewhat analogous to axes in MATLAB. The general syntax for minpages is as follows:

```
\begin{minipage}[pos][height][contentpos]{width}
```

Minipages are a somewhat advanced feature and we will not go into too much detail on the topic here yet they are used quite a bit in the `Slides_Template.tex` file that will be shared with you after the workshop.

Practice your whitespace management by trying to replicate Exercise4.pdf

6 Custom commands

When using commands that can be tedious to type (e.g. `\partial`), we can create new command handles that are easier to type out. For this one, we created a command such that `\del` also gives us the same symbol as `\partial`. At the beginning of the document, you can see where we typed this:

```
\newcommand{\del}{\partial}
```

So now we can retype Eqn. (6) using `\del` instead of `\partial` and we get the same result:

$$F(x) = \begin{cases} 0, & x \leq 0 \\ \dfrac{\partial u}{\partial y} + C_1^2 & x > 0 \end{cases}$$

Can you find the other custom commands that we made in this document?

7 Bibliographies & BibTeX

As we have seen in the previous sections, LaTeX is very very good at keeping track of the numbering of labels and compiling them in the proper order. This becomes EXTREMELY useful when it comes to creating a bibliography. In short, with the use of BibTeX a handle can be assigned to each reference. When you would like to cite the reference in writing all you need to do is use the command `\cite{·}`. LaTeX will take care of the rest!

7.1 Using BibTeX

Creating bibliographies in LaTeX is easy using BibTeX. BibTeX is awesome and a simple extension of LaTeX [1].

- Information for references is stored in a separate `.bib` file.
- For each reference the `.bib` file will include an entry with your label for that reference and other relevant citation information such as author, title, journal, year, publisher, etc...
- These `.bib` entries can usually be downloaded directly from the documents online source (eg., Google Scholar, Web of Knowledge, Elsevier, etc...) or generated with a third party literature management software such as Mendeley.
- Once the `.bib` file is populated with reference entries and compiled, a reference can be cited in the text using `\cite{ReferenceLabel}`. This will automatically add the reference to the bibliography and keep things in the order defined by the bibliography style being used.
- At the end of your document you will define the bibliography style using the command `\bibliographystyle{·}` and then point LaTeX to your BibTex file name using the command `\bibliography{Filename.bib}`
  - There are many different bibliography styles and many journals will have their own style defined which you can just insert in your manuscript writing. Here we are using IEEE Transactions which is a standard built in style in LaTeX.

Take a look at the code that generated the citation at the beginning of this subsection and the corresponding BibTeX file to see how it was done.

7.2 Creating a BibTeX File and Entry Content

Creating a `.bib` file is super easy. All you need to do is open a new file in LaTeX, include the references, and save it as a `.bib`. There is no limit to how many references you put in your `.bib` file and only the ones that are cited will show up in the reference section. There is no need for a document class definition at the top of your `.bib` file or any `\begin{·}` or `\end{·}` statements necessary. Like stated above, you can easily pull BibTeX citations from Google Scholar, but it is important to understand the content of a BibTeX entry. A sample reference entry for the `.bib` file is shown below.

```
@article{hawking1982,
title={The development of irregularities in a single bubble inflationary universe},
author={Hawking, Stephen W},
journal={Physics Letters B},
volume={115},
number={4},
pages={295--297},
year={1982},
publisher={Elsevier}
}
```

More information on different types of entries can be found here: `https://en.wikipedia.org/wiki/BibTeX`

Practice your BibTeX skills by trying to replicate Exercise5.pdf

8 Other Random Resources

- Templates - Can pull pre-generated style and class files for a specific document type (`http://www.latextemplates.com/`)
- Mathematica - Can copy inputs and outputs as LaTeX great for large tables and matrices
- Detexify - Can draw symbols and it will output the LaTeX command (`http://detexify.kirelabs.org/classify.html`)
- Codecogs - Real time typing of LaTeX math equations with a helpful user interface (`https://www.codecogs.com/latex/eqneditor.php`)

References

[1] A. A. McAuthorson, "An introduction to bibtex for ninjas and wizards," *J LaTeX*, vol. 007, no. 1, pp. 0–7, 1987.