tutorial-latex-document

Bryan Herman

January 2014

1 Creating Your First Document

- Explain sharelatex editing enviroment
- Explain error messages and debugging
- Choose a blank project
- Latex commands begins with a backslash
- Explain what a document class is, find a website with a list
- Explain packages
- Explain googling and CTAN
- Explain title, author date and what this really means
- Ends preamble
- Latex environments begin/end
- maketitle is a macro that sets up the title in the document
- Simple section introduction

2 Sections and Paragraphs

Above is how we make a section title.

2.1 This is a subsection

2.1.1 This is a subsubsection

If we want a something unnumbered we add a star *

This is a subsubsection title unnumbered

2.2 Paragraphs

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec vulputate orci. Pellentesque non metus sed odio rutrum bibendum. Mauris imperdiet luctus ipsum, eget euismod nisl cursus eu. Phasellus convallis justo vel tortor mollis, id semper dolor egestas. Integer pharetra eu turpis non tempor. Duis tellus enim, viverra in consequat eu, consectetur sed est. Etiam ac diam ut nulla vehicula porttitor quis vitae nisl. Fusce feugiat sollicitudin odio euismod varius. Nullam vehicula quam at leo tincidunt, sit amet iaculis diam imperdiet. This does not start a new paragraph. You need to leave an extra blank line like this.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc nec vulputate orci. Pellentesque non metus sed odio rutrum bibendum. Mauris imperdiet luctus ipsum, eget euismod nisl cursus eu. Phasellus convallis justo vel tortor mollis, id semper dolor egestas. Integer pharetra eu turpis non tempor. Duis tellus enim, viverra in consequat eu, consectetur sed est. Etiam ac diam ut nulla vehicula porttitor quis vitae nisl. Fusce feugiat sollicitudin odio euismod varius. Nullam vehicula quam at leo tincidunt, sit amet iaculis diam imperdiet.

Personally, I would rather have a blank line between paragraphs and not have an indent. Whenever you want to do something other than the default that Latex provides, simply search online and test it out. Here I will search in google "Latex no indent on paragraphs with spacing between". One website told me to use the package parskip. I added it at the top of the document and all looks good!

Finally, we will discuss commenting. To comment, just add a % in front of a line and it will not be seen by Latex, just for your eyes only.

By the way, the lipsum package is nice for fake text:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3 Fonts, Size, Color

There are many fonts you can choose when using latex. I usually choose a font that also has math fonts that look similar to the text fonts. First let's focus on font families. Here is a list of some font families I like to use:

- \usepackage[bitstream-charter]{mathdesign}
- \usepackage{palatino}
- \usepackage{fouriernc}

To change font sizes we can use the following commands:

- tiny – I am tiny text Need to surround by curly braces — latex scoping
- scriptsize – I am scriptsize text
- footnotesize – I am footnote size
- small – I am small size
- normalsize – Default
- large – I am large size
- Large – I am Large size
- huge – I am huge size
- Huge – I am Huge size

It should be noted that there are latex packages that give you more font sizes and custom font sizes. Just google! Now on to actual fonts. Here is a list of things you can do with a specific font family:

- shape
  - Upright - textup – I am upright text
  - Italics - textit – *I am italics text*
  - Small caps - textsc – I AM SMALL CAPS TEXT
  - Slanted - textsl – *I am slanted text*
  - Emphasize - emph – *Similar to italics (default), but behavior can change depending on packages*
- family
  - Roman family - textrm – I am Roman font
  - San serif text - textsf – I am san serif text
  - Typewriter text - texttt – `I am typewriter`
- weight
  - Bold - textbf – **I am bold**
  - Medium - textmd – I am medium
- misc
  - Use package `ulem` to get underlining and strikethrough capability
  - Underline - ul – I am underlined
  - Strikethrough - sout – ~~I have a line through me~~
  - There are many more options in the package documentation

Finally we discuss text coloring where we need the `xcolor`. For this we use latex scopes just like for fonts. Here is source code that makes some text blue and red.

```
{\color{blue}Let's start off blue} and {/color{red} switch to red}.
```

Let's start off blue and switch to red.

4 Changing the layout of the pages

4.1 Margins

In order to changes the margins, you need the geometry package. Just look it up online. We can use \usepackage[top=1in, bottom=1in, right=1in, left=1in]{geometry}. You can also change the paper size with this package as well. The default is US letter.

4.2 Page Styles

Page styles are a way to customize the theme of a page. This is mainly for headers and footers. The default page style is called `plain`. This includes just the page number at the bottom. It is set with the command \pagestyle{plain}. To remove the page style and have a blank header and footer just use \pagestyle{empty}. This is useful for title pages where the page number may be shown by default. If you use the above command you will see that all the following pages are also empty. To switch back you would have to figure out which text is on the second page and put the command back in for plain style. However, there is a slightly easier way.

Because we only want to change the page style on one page, we can use the command \thispagestyle{empty}. Now it will just change the style for the one page. Note, that the title page is sometimes special and you may have to use thispagestyle always.

4.2.1 FancyHDR

There is a nice package to control the headers and footer to make you own custom style. The package is called `fancyhdr`. Let's use it and see what it gives us. Because we are using the article class, the default behavior for this package is to put the first subsection of the page as the leftmark in the header and the last section title as the right mark. Let's say we want none of these and we just want the title to be in the center of the header, our name to be on the left footer and the page to be on the right footer. We can do the following:

```
\fancyhead{} % clears the header
\fancyfoot{} % clears the footer
\chead{tutorial-document-basic}
\lfoot{B. Herman}
\rfoot{\thepage/\pageref{LastPage}}
```

It should be noted that because I wanted to know the total number of pages, I used LastPage. This statement is part of the lastpage package and it has to be included. There is also a ton of more things that can be done. This includes predefinining multiple types of page styles, having even and odd customizations, taking into account book bindings, etc. Just search online!

5 Mathematics and Equations

Packages recommended for whenever you do any kind of equations: amsmath and amssymb.

5.1 Inline Equations

To do inline equations, we surround the equation by dollar signs: $2+3=5$. If you try to

5.2 Greek Letters

Greek letter are always important and Latex makes it easy to use them. Just put a backslash before the greek letter name. If you make the first letter captial you get the uppercase greek letter. Here are some examples - ϵ, α, ω, Ω. It should be noted that not all greek letters have an uppercase form because they are equivalent to English (e.g., Alpha (A), Beta (B)).

5.3 Math functions and symbols

There are an enormous amount of math functions. For example, all the trig you want - sin, cos, arctan. There is also exponential and log - exp, log. There are a bunch of symbols - $\leq$, $\equiv$, $\cdot$, $\in$. Just look them up online or you can use *detexify* where you can write the symbol you are looking for and it will come back and tell you the command and what latex package it is apart of.

One last item here is fractions. The command is `\frac{}{}`. In the first curly braces you but the numerator and for the last curly braces you but the denominator - $\frac{1}{2}$

5.4 Displayed Equations / Labels and Cross Referencing

Most of the time, you want you equation to be centered on its own line with an Equation number. This is no problem for latex when you use the equation environment.

$$2+3=5 \tag{1}$$

And another:

$$f(x)=a_0+\sum_{i=1}^{\infty}a_iP_i(x) \tag{2}$$

Now, as you can see, the equation number is automatically generated. Now if we want to reference this equation number in the text, we could just state Eq. (1). However, if we put another equation above this one it will no longer be Equation 1. We can use labels and cross referencing to have latex link a reference to an equation. To do this we must put a label before the end of an equation. It looks like - `\label{eq:expansion}`. It is good style to prefix the keyword expansion with the type of label it is: eq. We will have other prefixes for other label types later. Now that we added it to the expansion formula we can reference it with `\eqref` which will automatically surround it by parenthesis. In Eq. (2), we show a series expansion.

Finally, if you don't want your equation to have a number just add a * after equation in the begin and end commands.

$$3+3\neq 5$$

5.5 Equations on multiple lines

With very complex equations it is nice to break them up across multiple lines. The first example will use the `multline` environment. With this, we can easily break up the equation amoung multiple lines by just adding two backslashes -

$$\begin{multline}
\nabla\cdot\hat{\Omega}\varphi\left(\vec{r},E,\hat{\Omega}\right)+\Sigma_t(\vec{r},E)\varphi\left(\vec{r},E,\hat{\Omega}\right)= \\
+\int_0^{\infty}dE'\int_{4\pi}d\Omega'\sum_j\Sigma_s^j\left(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega}\right)\varphi\left(\vec{r},E',\hat{\Omega}'\right) \\
+\frac{1}{k_{eff}}\frac{1}{4\pi}\int_0^{\infty}dE'\sum_{j_f}\chi^{j_f}\left(E'\rightarrow E\right)\nu^{j_f}\left(E'\right)\Sigma_f^{j_f}\left(\vec{r},E'\right)\int_{4\pi}d\Omega'\varphi\left(\vec{r},E',\hat{\Omega}'\right) \tag{3}
\end{multline}$$

Sometimes, it is nice to have things line up. In the above equation, if we could line up all of the plus signs, the equation make look a little nicer. For this, we use the `align` environment. This basically works like a matrix which is described next. It basically makes the equation have as many rows a you want with two columns. Here we make the plus signs be the last item in the first column and the rest of the line in the second column. To separate columns we just add an & in the equation.

$$\begin{align}
\nabla\cdot\hat{\Omega}\varphi\left(\vec{r},E,\hat{\Omega}\right)+&\Sigma_t(\vec{r},E)\varphi\left(\vec{r},E,\hat{\Omega}\right)= \tag{4}\\
+&\int_0^{\infty}dE'\int_{4\pi}d\Omega'\sum_j\Sigma_s^j\left(\vec{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega}\right)\varphi\left(\vec{r},E',\hat{\Omega}'\right) \tag{5}\\
+&\frac{1}{k_{eff}}\frac{1}{4\pi}\int_0^{\infty}dE'\sum_{j_f}\chi^{j_f}\left(E'\rightarrow E\right)\nu^{j_f}\left(E'\right)\Sigma_f^{j_f}\left(\vec{r},E'\right)\int_{4\pi}d\Omega'\varphi\left(\vec{r},E',\hat{\Omega}'\right) \tag{6}
\end{align}$$

What you notice now is that this environment gives a number for each line. Let's say we just want an equation number for the last line - we put the `\nonumber` command on the other lines.

$$\begin{aligned} \overbrace{\nabla \cdot \hat{\Omega} \varphi\left(\vec{r}, E, \hat{\Omega}\right)}^{\text{Leakage Term}} + \Sigma_t (\vec{r}, E) \varphi\left(\vec{r}, E, \hat{\Omega}\right) = \nonumber \\ + \int_0^\infty dE' \int_{4\pi} d\Omega' \sum_j \Sigma_s^j \left(\vec{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega}\right) \varphi\left(\vec{r}, E', \hat{\Omega}'\right) \nonumber \\ + \frac{1}{k_{eff}} \frac{1}{4\pi} \int_0^\infty dE' \sum_{j_f} \chi^{j_f}\left(E' \rightarrow E\right) \nu^{j_f}\left(E'\right) \Sigma_f^{j_f}\left(\vec{r}, E'\right) \int_{4\pi} d\Omega' \varphi\left(\vec{r}, E', \hat{\Omega}'\right) \end{aligned} \tag{7}$$

5.6 Matrices

To build a matrix, we just use the matrix environment from within the equation environment. It works just like the align above. We just double backslash to separate rows and we use & to separate columns in a row.

$$\begin{matrix} 2 & 3 \\ 4 & 5 \end{matrix}$$

There are a bunch of modifications to the matrix environment depending on how you want to enclose it. We make an unordered list of these using the `itemize` environment where each bullet begins with the command `\item`:

- `pmatrix` environment - surrounds matrix by parenthesis
- `bmatrix` environment - surrounds matrix by square brackets
- `Bmatrix` environment - surrounds matrix by curly braces
- `vmatrix` environment - surrounds matrix by single vertical bars
- `Vmatrix` environment - surrounds matrix by double vertical bars

Here is an example for a determinant of a 2x2 matrix-

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

5.7 Accents

We all like to write in operator form so we dont have to write out Eq. (7) every time. Here are some options:

- `\mathcal{M}` - $\mathcal{M}$ - used for continuous operator
- `\mathbb{M}` - $\mathbb{M}$ - used for discretized operator / matrix
- `\mathbf{x}` - $\mathbf{x}$ - used for bolding possible for vectors
- `\boldsymbol{\Phi}` - $\boldsymbol{\Phi}$ - another way of bolding

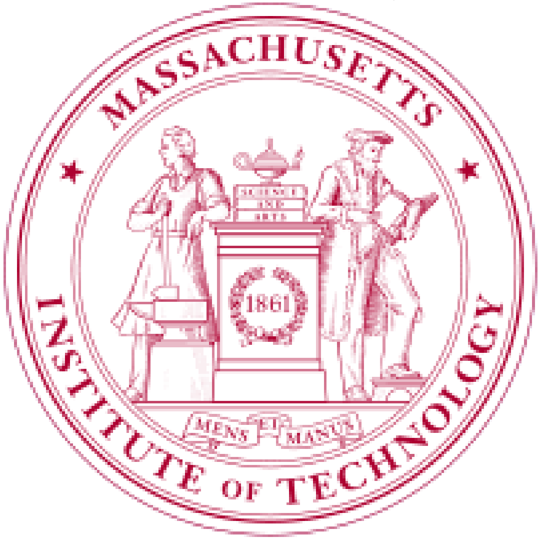
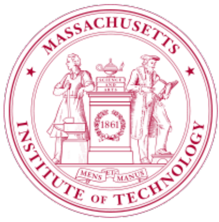
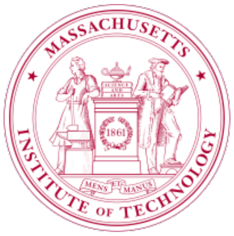
$$\mathbb{M}\Phi = \frac{1}{k}\mathbb{F}\Phi \tag{8}$$

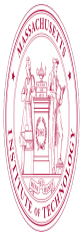
6 Figures, Subfigures and Tables

6.1 Figures

To include images into your document we must add the package `graphicx`. The command to include a graphic is `\includegraphics[]{}`. In the curly braces goes the relative path to the image and in the square brackets are options for the image. Some common options are -

- `scale=` - scale the image by some factor, keep aspect ratio
- `height=` - set the height of the image and keep the aspect ratio
- `width=` - set the width of the image and keep the aspect ratio
- `angle=` - rotate the image by some angle in degrees
- `clip=` - set this to true if you want to crop the image
- `trim=` - put in 4 values to crop the image. the order of the 4 values is left, bottom, right and top side.

It should be noted that if height and width are both used, the aspect ratio will not be kept constant unless those values keep that ratio. Let's play around with these by uploading the MIT seal image. Now let's include it -  As you see, it is just thrown into the text. We will discuss how to make this fit correctly in your document later, but let's first show the options. We can scale this down -  . We can set the height as some measure value. There are a ton of units in Latex. You could leave it as unitless, in for inches or cm for centimeters. Lets make the height 3cm - 

We could do the same thing for the width, but let's try both and see what happens to the aspect ratio - 

Now we can also rotate it 45 degrees - .

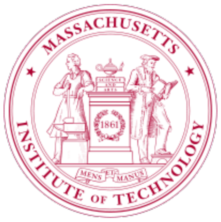
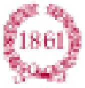


Figure 1. MIT Seal

Finally, let's say we just wanted the 1861 part of the image. We can first get it to the size we want and then crop around it. 

6.1.1 Floating Figures

Most of the time, you don't want to be just throwing you figure right between your text. If you want to include a nice figure with a caption that is displayed on a line by itself, we must put our image commands in a `figure` environment. This environment is also special because it is a *floating* environment. What this means is that Latex will move this figure as it sees fit to minimize extra whitespace. An example of this is below:

```
\begin{figure}[t,p]
\centering
\includegraphics[scale=0.4]{figs/MIT.png}
\caption{MIT Seal}
\label{fig:MITseal}
\end{figure}
```

After the environment begins, there are optional arguments [t,p] specified. This let's latex know your preferences on where the figure will be placed. If you include nothing, latex will do what it wants. Here we tell it that we would like this figure to be placed at the top of the page by putting at `t` first. Next, we put a `p` letting it know that if that doesn't work please put it on a separate page for floating environments. Other possibilities are bottom `b` and right here `h`. Also, if latex is not doing what you want, put the specifying that you want first followed by an ! to force it to do this.

Also we have included the command to center the image on the page called centering. Then follows the same includegraphics commands like before. To add a caption we use the caption command. Like equations, we put a label just before the end of the environment for cross referencing. This time we used the prefix fig.

Let's now put the figure code in right below this text: As you see, the figure went to the top of the page just like we preferred. Let's now put the same figure code in, but now have it show up right below this text: I don't really like the default style of the captions. Say I prefer that the Figure and number be bold, followed by a period and then my caption title. To do this we can use the caption package and include with some optional arguments for making the label boldface and the separator a period - `\usepackage[labelfont=bf, labelsep=period]{caption}`

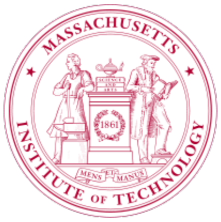


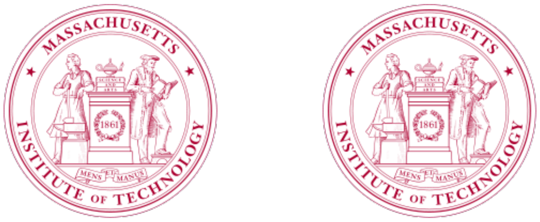
Figure 2. MIT Seal

6.2 Subfigures

If you want subfigures, you must include the package `subfigure`. Now we first make a figure environment and then put the subfigures in. It goes like this -

```
\begin{figure}[h]
\centering
\subfigure[Type 1]{\includegraphics[scale=0.4]{figs/MIT.png}\label{fig:MITsealsub1}}
\qquad\qquad % this just puts spacing between the figures
\subfigure[Type 2]{\includegraphics[scale=0.4]{figs/MIT.png}\label{fig:MITsealsub}\label{fig:MITsealsub2
\caption{MIT Seals}
\label{fig:MITseal3}
\end{figure}
```

Here it is displayed on the page -



(a) Type 1

(b) Type 2

Figure 3. MIT Seals

You can also cross reference subfigures as well as the whole figure environment. To cross reference things other than equations we just use `\ref`. In Figure 3 exists subfigures 3(a) and 3(b).

6.3 Tables

We have already talked about how to make new rows and columns from matrices. Here, we introduce the environment for tables called `tabular`. The structure of this environment is:

```
\begin{tabular}{l c c}
\hline
Fruit & At home & At work \\
\hline
\hline
Apples & 5 & 2 \\
\hline
Bananas & 3 & 1 \\
\hline
\end{tabular}
```

What you will notice with this structure is that after the begin command there is an extra curly brace needed for input. Unlike matrices, we can specify the position at which the text appears in a cell. Here, the first column is left justified and the rest are centered. The other two options are `r` for right justified and `p{3cm}` for a column of a certain size. To get horizontal lines we use the command hline. Let's see what this table looks like -

Fruit	At home	At work
Apples	5	2
Bananas	3	1

Sometimes you may want the columns to be separated by a vertical bar. To do this you add bars in the format specifier after begin like this -

```
\begin{tabular}{l|c|c}
\hline
Fruit & At home & At work \\
\hline
\hline
Apples & 5 & 2 \\
\hline
Bananas & 3 & 1 \\
\hline
\end{tabular}
```

Fruit	At home	At work
Apples	5	2
Bananas	3	1

Just as with figures, there is a floating environment for tables called `table`. We use it just like for figures. Here we use the prefix `tab` for the label to cross reference Table 1.

Table 1. Where is my fruit?

Fruit	At home	At work
Apples	5	2
Bananas	3	1

There is another way of putting together a table which has a more professional look. It is called book tables in the package booktabs. The differences in constructing this nicer table are:

```
\begin{tabular}{l c c}
\toprule
Fruit & At home & At work \\
\midrule
\midrule
Apples & 5 & 2 \\
\midrule
Bananas & 3 & 1 \\
\bottomrule
\end{tabular}
```

Table 2. Where is my fruit?

Fruit	At home	At work
Apples	5	2
Bananas	3	1

7 Generating Bibliographies with Bibtex

All bibliographic entries are placed in the .bib file. Although there are programs that can put these together for you, it is easy to just build it straight in the text file. To have the bibliography show up at the end of the document you put the following two commands:

```
\bibliographystyle{plain}
\bibliography{ref}
```

The first thing you need is a bibliography style which you can look up online. Here we just use plain. The second entry is the bibliography file itself without the .bib extension. We just place these as the last thing in the document before the end document command.

Let's look at the .bib file. If you look online, you will see all the possibilities for bibtex entries. With each type of entry, there are mandatory and optional fields. The most common you may use are inproceedings, book, article, PhDthesis. Note that conferences and journals will give you specific instructions on how to cite things. In each bibtex entry there is a key that is used for referencing in the text. To cite Paul Romano's 2013 OpenMC paper we just do `\cite{Romano2013_openmc}` [1]. Now say we want to site two things, we can comma separate them `\cite{Aviles2013,Horelik2013}` [2, 3]. Here we put, 3 in comma separated [4, 5, 6]. As you will see, the plain style doesn't order the references in numerical order. We use a package called natbib which gives us more flexibility. By default it will do author year styles, but we can do numerical referencing with:

```
\usepackage[numbers]{natbib}
```

For the bibliography style we can use `unsrtnat`.

References

[1] Paul K. Romano and Benoit Forget. The OpenMC Monte Carlo particle transport code. *Ann. Nucl. Energy*, 51:274 – 281, 2013.

[2] Daniel J. Kelly, III, Brian N. Aviles, and Bryan R. Herman. MC21 analysis of the MIT PWR benchmark: Hot zero power results. In *Mathematics and Computation 2013*, Sun Valley, ID, USA, 2013.

[3] Nick E. Horelik, Bryan R. Herman, Benoit Forget, and Kord S. Smith. Benchmark for evaluation and validation of reactor simulations (BEAVRS). In *Mathematics and Computation 2013*, Sun Valley, ID, USA, 2013.

[4] D. Lee. Convergence analysis of coarse mesh finite difference method applied to two-group three-dimensional neutron diffusion problem. *J. Nucl. Sci. Technol.*, 49(9):926–936, 2012.

[5] Paul K. Romano and Benoit Forget. Parallel fission bank algorithms in Monte Carlo criticality calculations. *Nucl. Sci. Eng.*, 170:125 – 135, 2012.

[6] Bryan R. Herman, Benoit Forget, and Kord S. Smith. Utilizing CMFD in OpenMC to estimate dominance ratio and adjoint. 2013.